Product: CylancePROTECT® and CylanceOPTICS

Document: Cylance API Guide. This guide is a succinct resource for analysts, administrators, and customers who are reviewing or evaluating the product.

Document Release Date: v2.0 rev24, February 2020

About BlackBerry Cylance®: BlackBerry Cylance develops artificial intelligence to deliver prevention-first, predictive security products and smart, simple, secure solutions that change how organizations approach endpoint security. BlackBerry Cylance provides full-spectrum predictive threat prevention and visibility across the enterprise to combat the most notorious and advanced cybersecurity attacks, fortifying endpoints to promote security hygiene in the security operations center, throughout global networks, and even on employees’ home networks. With AI-based malware prevention, threat hunting, automated detection and response, and expert security services, BlackBerry Cylance protects the endpoint without increasing staff workload or costs.

Copyright: © 2020 BlackBerry Cylance Inc. All Rights Reserved.

Global Headquarters
400 Irvine Spectrum Drive, Irvine, CA 92618

Professional Services Hotline
+1-877-97DEFEND • +1-877-973-3336

Corporate Contact
+1-914-CYLANCE • +1-914-295-2623

Email
sales@cylance.com

Website
https://www.cylance.com

To Open a Support Ticket
https://support.cylance.com — Click on Submit a Ticket

To View Knowledge Base and Announcements
Login to https://support.cylance.com

To Request a Callback from BlackBerry Cylance Support
+1-866-699-9689
Table of Contents
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of Contents</td>
<td>3</td>
</tr>
<tr>
<td>Application Management</td>
<td>9</td>
</tr>
<tr>
<td>Application Management</td>
<td>10</td>
</tr>
<tr>
<td>To Add an Application</td>
<td>10</td>
</tr>
<tr>
<td>To Edit an Application</td>
<td>13</td>
</tr>
<tr>
<td>To Delete an Application</td>
<td>14</td>
</tr>
<tr>
<td>To Regenerate an Application Control</td>
<td>15</td>
</tr>
<tr>
<td>Copy Tenant ID</td>
<td>16</td>
</tr>
<tr>
<td>API in Audit Logs</td>
<td>16</td>
</tr>
<tr>
<td>RESTful API</td>
<td>18</td>
</tr>
<tr>
<td>Authentication and Authorization</td>
<td>19</td>
</tr>
<tr>
<td>Application</td>
<td>19</td>
</tr>
<tr>
<td>Service Endpoint</td>
<td>19</td>
</tr>
<tr>
<td>Authentication</td>
<td>20</td>
</tr>
<tr>
<td>Authentication Token</td>
<td>20</td>
</tr>
<tr>
<td>Generating the Authentication and Access Tokens</td>
<td>21</td>
</tr>
<tr>
<td>Token Lifecycle</td>
<td>23</td>
</tr>
<tr>
<td>Request / Response Model</td>
<td>23</td>
</tr>
<tr>
<td>Authorization</td>
<td>24</td>
</tr>
<tr>
<td>Access Token</td>
<td>24</td>
</tr>
<tr>
<td>Response Status Codes</td>
<td>24</td>
</tr>
<tr>
<td>About Device ID</td>
<td>25</td>
</tr>
<tr>
<td>About Zone ID</td>
<td>26</td>
</tr>
<tr>
<td>User API</td>
<td>27</td>
</tr>
<tr>
<td>Create User</td>
<td>28</td>
</tr>
<tr>
<td>Get Users</td>
<td>31</td>
</tr>
<tr>
<td>Get User</td>
<td>34</td>
</tr>
<tr>
<td>Update User</td>
<td>36</td>
</tr>
<tr>
<td>Delete User</td>
<td>38</td>
</tr>
<tr>
<td>Send Invite Email</td>
<td>39</td>
</tr>
<tr>
<td>Send Request Password Email</td>
<td>41</td>
</tr>
<tr>
<td>Device API</td>
<td>42</td>
</tr>
<tr>
<td>Get Devices</td>
<td>43</td>
</tr>
<tr>
<td>Feature</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Get Memory Protection Event</td>
<td>160</td>
</tr>
<tr>
<td>Memory Violation Types</td>
<td>162</td>
</tr>
<tr>
<td><strong>Detection API</strong></td>
<td>165</td>
</tr>
<tr>
<td>Get Detections</td>
<td>166</td>
</tr>
<tr>
<td>Get Detection</td>
<td>169</td>
</tr>
<tr>
<td>Get Recent Detections</td>
<td>176</td>
</tr>
<tr>
<td>Get Detections CSV</td>
<td>177</td>
</tr>
<tr>
<td>Get Detections By Severity</td>
<td>180</td>
</tr>
<tr>
<td>Update Detection</td>
<td>182</td>
</tr>
<tr>
<td>Delete Detection</td>
<td>184</td>
</tr>
<tr>
<td>Delete Detections</td>
<td>185</td>
</tr>
<tr>
<td><strong>Package Deployment API</strong></td>
<td>186</td>
</tr>
<tr>
<td>Create Package</td>
<td>187</td>
</tr>
<tr>
<td>Get Packages</td>
<td>189</td>
</tr>
<tr>
<td>Get Package</td>
<td>193</td>
</tr>
<tr>
<td>Delete Package</td>
<td>195</td>
</tr>
<tr>
<td>Create Package Execution</td>
<td>196</td>
</tr>
<tr>
<td>Get Package Executions</td>
<td>200</td>
</tr>
<tr>
<td>Get Package Execution</td>
<td>204</td>
</tr>
<tr>
<td>Delete Package Execution</td>
<td>206</td>
</tr>
<tr>
<td><strong>Detection Rule API</strong></td>
<td>209</td>
</tr>
<tr>
<td>Get Detection Rule List</td>
<td>210</td>
</tr>
<tr>
<td>Get Detection Rule CSV List</td>
<td>213</td>
</tr>
<tr>
<td>Get Detection Rule</td>
<td>214</td>
</tr>
<tr>
<td>Validate Detection Rule</td>
<td>218</td>
</tr>
<tr>
<td>Create Detection Rule</td>
<td>221</td>
</tr>
<tr>
<td>Update Detection Rule</td>
<td>226</td>
</tr>
<tr>
<td>Deactivate or Delete Detection Rule</td>
<td>231</td>
</tr>
<tr>
<td>Get Detection Rule Natural Language Representation</td>
<td>232</td>
</tr>
<tr>
<td>Get Detection Rule Counts</td>
<td>233</td>
</tr>
<tr>
<td><strong>Detection Rule Sets</strong></td>
<td>235</td>
</tr>
<tr>
<td>Get Detection Rule Set List</td>
<td>236</td>
</tr>
<tr>
<td>Get Detection Rule Set CSV List</td>
<td>239</td>
</tr>
<tr>
<td>Get Detection Rule Set</td>
<td>240</td>
</tr>
</tbody>
</table>
Application Management

Cylance Console administrators can manage multiple API applications, including the access privileges to your Cylance Console data, CylancePROTECT and CylanceOPTICS.

To Add an Application

IMPORTANT: A tenant can have up to 10 Custom Applications.

1. Log in to the Cylance Console as an administrator. Only administrators can create an application integration.

2. Select Settings > Integrations.

3. Click Add Application.

4. Type an Application Name. This must be unique within your organization.

5. Select the access privileges for a Console data type. Not selecting any checkboxes for a data type means the application does not have access to that data type.
6. Click Save. The credentials to use for the application displays.

7. Copy and paste the Application ID and Application Secret to your API application. Or you can click OK to close the dialog box. You can view the Application ID and Application Secret from the Integrations page.

**Note:** There are some API operations listed in the Add Application matrix that can be enabled (Global List - Read and Modify; Policy - Write, Modify, and Delete) but are not available with the initial release. These API operations are currently under development and will be available in a future release.
<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CylanceOPTICS Commands</td>
<td>The CylanceOPTICS Device Commands. Includes Device Lockdown (locking a device, retrieving history) and File Retrieval (requesting, checking, and getting results).</td>
</tr>
<tr>
<td>CylanceOPTICS Detections</td>
<td>The CylanceOPTICS Detection Events triggered by the Context Analysis Engine (CAE). Enables further automation of analyzing, triaging, and responding to malicious or suspicious activity prevented or detected by CylanceOPTICS.</td>
</tr>
<tr>
<td>CylanceOPTICS Exceptions</td>
<td>The CylanceOPTICS Detection Exceptions that adds exceptions to Detection Rules.</td>
</tr>
<tr>
<td>CylanceOPTICS Focus Views</td>
<td>The CylanceOPTICS Focus Views retrieves an information trail starting with the first event related to an artifact from an InstaQuery result or CylancePROTECT event.</td>
</tr>
<tr>
<td>CylanceOPTICS InstaQueries</td>
<td>The CylanceOPTICS InstaQuery allows searching for system artifacts stored locally by CylanceOPTICS - files, registry key persistence points, processes, etc.</td>
</tr>
<tr>
<td>CylanceOPTICS Policies</td>
<td>The CylanceOPTICS settings in a Policy. This requires the Policy settings to also be enabled.</td>
</tr>
<tr>
<td>CylanceOPTICS Rule Sets</td>
<td>The CylanceOPTICS set of rules that is then applied to a Policy.</td>
</tr>
<tr>
<td>CylanceOPTICS Rules</td>
<td>The CylanceOPTICS Detection Rules that help monitor an organization for security threats or anomalous behavior.</td>
</tr>
<tr>
<td>Devices</td>
<td>Devices are systems with a Cylance Agent installed. You can get information about devices in your organization. You can also update (Modify) or remove (Delete) devices from your organization.</td>
</tr>
<tr>
<td>Global List</td>
<td>Global Lists include the Safe List and the Global Quarantine List. Each Global List operation has its own set of required and optional request fields.</td>
</tr>
<tr>
<td>Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Packages Configuration</td>
<td>The CylanceOPTICS packages to be sent and stored on devices. Note: CylanceOPTICS packages are not sent to devices by default. Devices must receive a command to download a package.</td>
</tr>
<tr>
<td>Packages Deployment</td>
<td>The CylanceOPTICS packages to execute on devices.</td>
</tr>
<tr>
<td>Policies</td>
<td>Policies contain the protection settings applied to devices. Policies allow adding and removing devices instead of needing to manually update each device when you want to change the protection settings.</td>
</tr>
<tr>
<td>Threats</td>
<td>Threat Details provide information about a file as well as reference information about why a file is considered Safe or a Threat. Use the Threats request to get this information.</td>
</tr>
<tr>
<td>Users</td>
<td>Users have access to the data in the Console, based on the role assigned to the User. For example, an Administrator can see everything in the Console, while a User is limited to the zones to which the User is assigned.</td>
</tr>
<tr>
<td>Zones</td>
<td>Each device belongs to at least one Zone. Zones are similar to tags and assist in organizing your devices.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete</td>
<td>Ability to delete the data.</td>
</tr>
<tr>
<td>Modify</td>
<td>Ability to modify existing data. Also known as update.</td>
</tr>
<tr>
<td>Read</td>
<td>Ability to read data but cannot create, modify, or delete the data. Also known as list.</td>
</tr>
<tr>
<td>Write</td>
<td>Ability to add data to the Console. Also known as create.</td>
</tr>
</tbody>
</table>

**To Edit an Application**

1. Log in to the Cylance Console as an administrator. Only administrators can edit an application integration.
2. Select **Settings > Integrations**.
3. Click the Edit icon for the application you want to change.
4. Edit the privileges, then click **Save Changes**.

**Figure 3: Edit Custom Application**

To Delete an Application

1. Log in to the Cylance Console as an administrator. Only administrators can delete an application integration.

2. Select **Settings > Integrations**.

3. Click the Remove icon for the application you want to remove.

**Figure 4: Save Custom Application**

**Figure 5: Remove Custom Application**

4. Click **Remove Application** to confirm the deletion.
To Regenerate an Application Control

There may be times when it is necessary to regenerate the credentials for an Integration, like when credentials are compromised or stolen. For Cylance Integrations, regenerating credentials creates a new Application Secret; the Application ID remains unchanged.

**Note:** After regenerating credentials, you must update this information in the application used to generate the API access token.

1. Log in to the Cylance Console as an administrator. Only administrators can regenerate an application credential.

2. Select **Settings > Integrations**.

3. Click the down arrow to expand the information for the application for which you want to regenerate credentials.

4. Click **Regenerate Credentials**. A confirmation message appears.

5. Click **Yes, Regenerate** to confirm regenerating the credentials.
Copy Tenant ID

The Tenant ID is required for authorization. You can copy your Tenant ID from the Integrations page.

API in Audit Logs

The API calls listed below are included in the Console Audit Log (My Account > Audit Log) when something is created or updated. In the Audit Log, the "Who" field displays the Application Name, not the username.

- **Policy**: Create, Update, or Delete
- **Global List**: Add or Delete
- **Zone**: Create, Update, or Delete
- **Tenant User**: Create, Update, or Delete
- **Device**: Update Device, Update Device Threat, or Delete Device
RESTful API
Cylance provides RESTful APIs for registered organizations to manage their resources. To access the Cylance API resources, the client will need to follow the authentication and authorization flow as defined below. This requires the client to send a request to the Auth endpoint, which will return an access token that the client will use for calling all other endpoints.

**Note:** Cylance supports Cylance API resources, including helping users troubleshoot Cylance API requests. Cylance does not write or train users on how to create scripts or code (like using Python).

### Authentication and Authorization

#### Application

An Application acts as an integration point between the client system and the Cylance API. Through the Application, the client system is granted temporary access to act upon resources. Actions will be limited by the scopes associated to the Application itself, as defined in the "Application Management" on page 10 section.

#### Service Endpoint

The service endpoint address can contain a region code to identify the set of servers to which your organization belongs.

**Example for Europe - Central:** http://protectapi-euc1.cylance.com/devices/v2

**Note:** North America and US Government have a different format. See Service Endpoint column below for examples.

<table>
<thead>
<tr>
<th>Region</th>
<th>Code</th>
<th>Service Endpoint with Region Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia-Pacific - North</td>
<td>apne1</td>
<td><a href="https://protectapi-apne1.cylance.com/">https://protectapi-apne1.cylance.com/</a></td>
</tr>
<tr>
<td>Asia-Pacific - Southeast</td>
<td>au</td>
<td><a href="https://protectapi-au.cylance.com/">https://protectapi-au.cylance.com/</a></td>
</tr>
<tr>
<td>Europe - Central</td>
<td>euc1</td>
<td><a href="https://protectapi-euc1.cylance.com/">https://protectapi-euc1.cylance.com/</a></td>
</tr>
<tr>
<td>North America</td>
<td></td>
<td><a href="https://protectapi.cylance.com/">https://protectapi.cylance.com/</a></td>
</tr>
<tr>
<td>South America</td>
<td>sae1</td>
<td><a href="https://protectapi-sae1.cylance.com/">https://protectapi-sae1.cylance.com/</a></td>
</tr>
<tr>
<td>US Government</td>
<td>us</td>
<td><a href="https://protectapi.us.cylance.com/">https://protectapi.us.cylance.com/</a></td>
</tr>
</tbody>
</table>
Authentication

During the step which a client system requests access prior to using Cylance Resources, there is an independent Web API that will handle the Authentication process and grant access to the client system. A token based authentication approach is being taken as a means of data transportation between the parties. Cylance has adopted JWT (RFC 7519) as the token format for its simplicity as well as its capabilities for digital signature.

The following actors exist in the Authentication workflow:

- **Authentication Token**: Created and signed by the client system to perform an Authentication request, it is in this request where the Application is indicated.

- **Authentication Endpoint**: Part of the Cylance Auth Web API which will handle the authentication requests coming from client systems, there will be a particular endpoint to handle JWT tokens.

- **Access Token**: If authentication is successful and the client system is granted access to the requested application, a token representing this identity and some key attributes will be returned as a JWT token.

Authentication Token

The Authentication Token contains the ID of the Application to which a client system is requesting access. The Application contains two attributes: **Application ID** and **Application Secret**, the latter is cryptographic nonce used to sign the token, thus ensuring the authenticity of the caller and therefore, it must be shared between client and server. The Authentication endpoint has a mechanism to verify the signature and eventually proceed to grant access to the Application, if the client request is indeed allowed.

The client will create the Authentication token by indicating the Application ID as a claim and sign it using the Application Secret.

The Authentication Token must have the following claims. All are registered and conform to the JWT standard.

<table>
<thead>
<tr>
<th>Claim</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| exp   | NumericDate| Date and time when the Token expires and is no longer valid for processing. This is Unix epoch time in seconds. **Note**: The longest time-span honored by the service is 30 minutes from the value specified in the iat claim. Specifying a longer time-span will result in an HTTP 400 (Bad Request) response from the server.
**Claim** | **Type** | **Description**
--- | --- | ---
iat | NumericDate | Time when the token was issued. This is Unix epoch time in seconds.
isss | StringOrUri | Represents the principal issuing the token, which is [http://cylance.com](http://cylance.com).
jti | String | Unique ID for the token, which can be used to prevent reply attacks.
sub | StringOrUri | Principal subject to the claim. In our case, this would hold our Application ID.

**Custom Claims**

| scp | String | Comma delimited scopes requested. This claim is optional. For a list of scopes and descriptions, see ["Scope Values for Authentication Token" on page 331](#).

tid | String | Tenant ID (available on the Integrations page in the Console).

**Example:**

```csharp
DateTime now = DateTime.UtcNow;
long unixTimestamp = now.ToUnixTimestamp();

token.Claims.Add("iat", now.ToUnixTimestamp());
token.Claims.Add("exp", now.AddMinutes(1).ToUnixTimestamp());
token.Claims.Add("sub", "k45f6798092hjdhs836h");
token.Claims.Add("jti", "k45f6798092hjdhs836h+d82c7976-ef46-47b6-80ce-4dda3c91bba3");
token.Claims.Add("tid", "f00e9987-ee61-57b7-80cf-5ebe3d02cbb4");
```

**Generating the Authentication and Access Tokens**

The Authentication Token can be generated using Python. You can use the Python example below, adding the required token claims that you need.

**Note:** Cylance Support does not provide assistance with installing non-Cylance programs (like Python) or security (like PyJWT). Cylance does have a knowledge base article with an example for installing [Python and PyJWT on Windows](#); this example is provided AS-IS and there is no guarantee the example will work in your environment.

**Software Requirements:**
- Python 3.7 (latest version recommended)
- PyJWT package (pip install PyJWT)
- Requests package (pip install requests)

**Notes:**
- Copying the Python Example from the PDF requires proper formatting in Python due to the extra line breaks that can cause an error. Use the Python and PyJWT on Windows knowledge base article to copy the example.
- Example using C# are available upon request.

**Python Example**

```python
# WARNING: Copying this example from the PDF requires proper
# formatting in Python due to the extra lines breaks that
# can cause an error.
# RECOMMENDED: Use the Python and PyJWT on Windows
# article to copy the example.

import jwt # PyJWT version 1.7.1 as of the time of authoring.
import uuid
import requests # requests version 2.22.0 as of the time of authoring
import json
from datetime import datetime, timedelta

# 30 minutes from now
timeout = 1800
now = datetime.utcnow()
timeout_datetime = now + timedelta(seconds=timeout)
epoch_time = int((now - datetime(1970, 1, 1)).total_seconds())
epoch_timeout = int((timeout_datetime - datetime(1970, 1, 1)).total_seconds())

jti_val = str(uuid.uuid4())
tid_val = "" # The tenant's unique identifier.
app_id = "" # The application's unique identifier.
app_secret = "" # The application's secret to sign the auth token with.

AUTH_URL = "https://protectapi.cylance.com/auth/v2/token"

claims = {
    "exp": epoch_timeout,
    "iat": epoch_time,
    "iss": "http://cylance.com",
    "sub": app_id,
    "tid": tid_val,
    "jti": jti_val
# The following is optional and is being noted here as an example
```
Python Example

```python
on how one can restrict
    # the list of scopes being requested
    # "scp":
    ["policy:create","policy:list","policy:read","policy:update"]

encoded = jwt.encode(claims, app_secret, algorithm='HS256').decode
    ('utf-8')
print ('auth_token:
    ' + encoded + "

payload = {"auth_token": encoded}
headers = {"Content-Type": "application/json; charset=utf-8"}
resp = requests.post(AUTH_URL, headers=headers, data=json.dumps
    (payload))

print("http_status_code: " + str(resp.status_code))
print("access_token:\n" + json.loads(resp.text)['access_token'] + "\n")
```

Token Lifecycle

An Authentication token should be used only once per request. This means the same token should not be usable for more than one request to prevent impersonation attempts. The jti attribute uniquely identifies the token. It can be used to keep track of all the tokens and prevent them from being reused. To ensure that the authentication token can be used only once, an expiration is enforced on the token. This means the token is usable within a few minutes or less.

Request / Response Model

<table>
<thead>
<tr>
<th>HTTP Method</th>
<th>POST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint</td>
<td>/auth/v2/token</td>
</tr>
<tr>
<td>Example</td>
<td><a href="https://protectapi.cylance.com/auth/v2/token">https://protectapi.cylance.com/auth/v2/token</a></td>
</tr>
<tr>
<td>Request Headers</td>
<td>Accept: application/json</td>
</tr>
<tr>
<td></td>
<td>Content-Type: application/json</td>
</tr>
<tr>
<td></td>
<td>Authorization: Bearer &lt;JWT Token returned by Auth API&gt; with the user:create scope encoded.</td>
</tr>
</tbody>
</table>
| Request     | {
|             | "title": "Authorization Request",
|             | "type": "object"
|             | "properties": {
|             | "auth_token": { |
Authorization

In response to the Authentication request, the client will receive a response that contains at least the Access Token. The access token will contain the Scopes that will dictate what can or cannot be done. This token is signed by the server and the client will merely echo it on every request as it tries to access Resources.

The access token represents the identity of the requester as well as some attributes like Scopes. This token will have an expiration and should be sent on every request in the Authorization Request Header. Failing to do so will result in an HTTP/1.1 401 Unauthorized response. Should the token be provided and prove to be legitimate but the server finds the action the caller is trying to attempt is not allowed (found in the scopes granted), an HTTP/1.1 403 Forbidden will be returned.

Access Token

The Access token represents a grant to access Cylance Resources. It contains information about the identity of the caller (Application) as well as control information form the Token itself, for instance, date it was issued and expiration. This token is also responsible for holding all Scopes that would be used by our system to validate actions attempted to be taken against Cylance Resources.

There is an expiration associated to this Token. The expiration time will be set during token creation on the server side. After the token expires, the server will respond with HTTP/1.1 401 Unauthorized indicating to the caller to authenticate again with a new access token.

Response Status Codes

Each API request will receive a response with a JSON payload and a standard HTTP status code.

Note: Some API request sections include additional response status descriptions (specific to that request) to help you troubleshoot issues.
<table>
<thead>
<tr>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 - OK</td>
<td>A successful call and operation. The response payload will be JSON, structured according to the nature of the request.</td>
</tr>
<tr>
<td>400 - Bad Request</td>
<td>There was a problem with the structure of the request or the payload. If determinable, the response payload will identify the failure in the request. A common case of this type of error is malformed JSON in the request body. A JSON validator can be used to troubleshoot these issues.</td>
</tr>
<tr>
<td>401 - Unauthorized</td>
<td>Invalid credentials were passed or some other failure in authentication.</td>
</tr>
<tr>
<td>403 - Forbidden</td>
<td>Request has been successfully authenticated, but authorization to access the requested resource was not granted.</td>
</tr>
<tr>
<td>404 - Not Found</td>
<td>A request was made for a resource that doesn’t exist. Common causes are either an improperly formed URL or an invalid API key.</td>
</tr>
<tr>
<td>409 - Conflict</td>
<td>A request was made to create or update an aspect of the resource that conflicts with another. The most common reason for this code is a Tenant name or User email that is already in use.</td>
</tr>
<tr>
<td>500 - Internal Server Error</td>
<td>A catch-all code response for any unhandled error that has occurred on the server. Contact Cylance Support for help with this issue.</td>
</tr>
<tr>
<td>501 - Not Implemented</td>
<td>A request was made against a resource with an operation that has yet to be implemented. Such operations should be identified accordingly in documentation.</td>
</tr>
<tr>
<td>Other</td>
<td>Contact Cylance Support if you encounter any status codes that are not on this list.</td>
</tr>
</tbody>
</table>

**About Device ID**

When attempting to query a CylanceOPTICS API call that utilizes a device ID value, be aware of the following:

The format for the CylanceOPTICS API device ID value should be:

**CylanceOPTICS Example:**

- 45E07F34E76B4A9EB167D6D0C510D6BA (upper case without dashes)

Passing the device ID value as the CylancePROTECT format will return an HTTP 200 status, as if the call was successful, but you will receive an incorrect response.

**CylancePROTECT Example:**

- 45e07f34-e76b-4a9e-b167-d6d0c510d6ba (lower case with dashes)
To obtain the device ID, you must query the CylancePROTECT API, then format the device ID to match the CylanceOPTICS format (see example above).

This query can be found in the Device API section of this document. Use the Get Devices and Get Device requests from the guide. The device ID value is the field titled "id".

**About Zone ID**

When attempting to query a CylanceOPTICS API call that utilizes a zone ID value, be aware of the following:

The format for the CylanceOPTICS API zone ID value should be:

**CylanceOPTICS Example:**

- D27FF5C45C0D4F56A00DA1FB297E440E (upper case without dashes)

Passing the zone ID value as the CylancePROTECT format will return an HTTP 200 status, as if the call was successful, but you will receive an incorrect response.

**CylancePROTECT Example:**

- d27ff5c4-5c0d-4f56-a00d-a1fb297e440e (lower case with dashes)

To obtain the zone ID, you must query the CylancePROTECT API, then format the zone ID to match the CylanceOPTICS format (see example above).

This query can be found in the Zone API section of this document. Use the Get Zones and Get Zone requests from the guide.
User API
Create User

Create (add) a new Console user. This requires a unique email address for the user being created.

Service Endpoint:

- /users/v2

Example: https://protectapi.cylance.com/users/v2

Method:

- HTTP/1.1 POST

Request Header:

- Accept: application/json
- Content-Type: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the user:create scope encoded.

Request:

```json
{
  "email": "testuser@email.com",
  "user_role": "00000000-0000-0000-0000-000000000001",
  "first_name": "Test",
  "last_name": "User",
  "zones": [
    {
      "id": "d27ff5c4-5c0d-4f56-a00d-a1fb297e440e",
      "role_type": "00000000-0000-0000-0000-000000000002"
    }
  ]
}
```

Response:

201 Created

```json
{
  "id": "a2c0ac7a-a63d-4583-b646-ae10db9c9768",
  "tenant_id": "4b1640d2-d563-41cf-94a7-0da1dca6aa98",
  "first_name": "Test",
  "last_name": "User",
  "email": "testuser@email.com",
  "has_logged_in": false,
  "role_type": "00000000-0000-0000-0000-000000000001",
  "role_name": "User",
}
```
"default_zone_role_type": "00000000-0000-0000-0000-000000000000",
"default_zone_role_name": "None",
"zones": [
  {
    "id": "d27ff5c4-5c0d-4f56-a00d-alfb297e440e",
    "role_type": "00000000-0000-0000-0000-000000000002"
  }
]
"date_last_login": null,
"date_email_confirmed": null,
"date_modified": "2019-09-13T22:33:26.098Z"
)

400 Bad Request - Returned for the following reasons:

- The User create request was empty.
- The Tenant ID cannot be retrieved from the JWT Token.
- The User's email address specified is not a proper email address.
- The User application role specified is not one of the accepted values.
- The zones array is empty when the User application role is not Administrator.
- The email provided is already in use.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

500 Internal Server Error - An unforeseeable error has occurred.

Request JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>email</td>
<td>The user's email address. This must be unique.</td>
</tr>
<tr>
<td>first_name</td>
<td>The user's first name. Maximum of 64 characters.</td>
</tr>
<tr>
<td>last_name</td>
<td>The user's last name. Maximum of 64 characters.</td>
</tr>
<tr>
<td>user_role</td>
<td>The user's role in the Console.</td>
</tr>
<tr>
<td></td>
<td>- User: 00000000-0000-0000-0000-000000000001</td>
</tr>
<tr>
<td></td>
<td>- Administrator: 00000000-0000-0000-0000-000000000002</td>
</tr>
<tr>
<td></td>
<td>- Read-Only: 00000000-0000-0000-0000-000000000003</td>
</tr>
<tr>
<td>zones</td>
<td>The zones to which the user has access. This is an array of elements.</td>
</tr>
<tr>
<td></td>
<td>- id: The unique identifier for the zone.</td>
</tr>
<tr>
<td></td>
<td>- role_type: The user's role for this particular zone.</td>
</tr>
</tbody>
</table>
**Field Name** | **Description**
---|---
Zone Manager: 00000000-0000-0000-0000-000000000001  
User: 00000000-0000-0000-0000-000000000002  
**Note:** If the user is an Administrator, the zones array is not required.

**Note:** To create a Zone Manager, set the user_role to User and assign a zone or zones to the User via the Zones parameter. Setting the user_role to Read-Only and using the Zones parameter will result in a bad request error.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date_created</td>
<td>The date and time (in UTC) the Console user was created.</td>
</tr>
<tr>
<td>date_email_confirmed</td>
<td>The date and time (in UTC) when the user confirmed the email provided. This should be null because the user account was recently created.</td>
</tr>
<tr>
<td>date_last_login</td>
<td>The date and time (in UTC) the user last logged in to the Console. This should be null because the user account was recently created.</td>
</tr>
<tr>
<td>date_modified</td>
<td>The date and time (in UTC) the Console user information was last updated.</td>
</tr>
<tr>
<td>default_zone_role_name</td>
<td>The name of the role for the user in the zone.</td>
</tr>
</tbody>
</table>
| default_zone_role_type | The unique identifier for the user's default role when assigned to a zone.  
- None: 00000000-0000-0000-0000-000000000000  
- Zone Manager: 00000000-0000-0000-0000-000000000001  
- User: 00000000-0000-0000-0000-000000000002  
<p>| email | The user's email address. |
| first_name | The user's first name. |
| has_logged_in | This should be false because the user account was recently created. |
| id | The user's unique identifier for the Console. |
| last | The user's last name. |</p>
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the user's role in the Console.</td>
</tr>
<tr>
<td>role_name</td>
<td>The unique identifier defining the user's role in the Console.</td>
</tr>
</tbody>
</table>
| role_type | - User: 00000000-0000-0000-0000-000000000001  
- Administrator: 00000000-0000-0000-0000-000000000002  
- Read-Only: 00000000-0000-0000-0000-000000000003  
- Zone Manager: 00000000-0000-0000-0000-000000000004  
Note: To create a Zone Manager, set the user_role to User and assign a zone or zones to the User via the Zones parameter. Setting the user_role to Read-Only and using the Zones parameter will result in a bad request error. |
| zones | The zones to which the user has access. This is an array of elements. |
| id | The unique identifier for the zone. |
| role_type | The user's role for this particular zone. |
| role_name | The name of the user's role in the zone. |

Note: If the user is an Administrator, the zones array will display empty brackets [ ].

Get Users

Request a page with a list of Console user resources belonging to a tenant, sorted by the created date, in descending order (most recent user registered listed first). The page number and page size parameters are optional. When the values are not specified, the default values are 1 and 10 respectively. The maximum page size that can be specified is 200 entries per page.

Service Endpoint:

- /users/v2?page=m&page_size=n

Append the following optional query string parameters:

- page: The page number to request
- page_size: The number of device records to retrieve per page

For example, to return the first page with up to 100 users:

https://protectapi.cylance.com/users/v2?page=1&page_size=100
Method:
- HTTP/1.1 GET

Request Headers:
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the user:list scope encoded

Request:
None

Response:
200 OK

```json
{
    "page_number": "1",
    "page_size": "10",
    "total_pages": "1",
    "total_number_of_items": "1",
    "page_items": [
        {
            "id": "a2c0ac7a-a63d-4583-b646-ae10db9c9768",
            "tenant_id": "4b1640d2-d563-41cf-94a7-0da1dca6aa98",
            "first_name": "Test",
            "last_name": "User",
            "email": "testuser@email.com",
            "has_logged_in": "true",
            "role_type": "00000000-0000-0000-0000-000000000001",
            "role_name": "User",
            "default_zone_role_type": "00000000-0000-0000-0000-000000000000",
            "default_zone_role_name": "None",
            "zones": [
                {
                    "id": "d27ff5c4-5c0d-4f56-a00d-a1fb297e440e",
                    "role_type": "00000000-0000-0000-0000-000000000002",
                    "role_name": "User"
                }
            ]
        },
        "date_modified": "2019-09-13T22:33:26.098Z"
    ]
}
```

400 Bad Request - Returned for the following reasons:
- The Tenant ID cannot be retrieved from the JWT token
- The page number or page size specified is less than or equal to zero, or the page size is greater than 200

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
500 Internal Server Error - An unforeseeable error has occurred.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date_created</td>
<td>The date and time (in UTC) the Console user was created.</td>
</tr>
<tr>
<td>date_email_confirmed</td>
<td>The date and time (in UTC) when the user confirmed the email provided.</td>
</tr>
<tr>
<td>date_last_login</td>
<td>The date and time (in UTC) the user last logged in to the Console.</td>
</tr>
<tr>
<td>date_modified</td>
<td>The date and time (in UTC) the Console user information was last updated.</td>
</tr>
<tr>
<td>default_zone_role_name</td>
<td>The name of the role for the user in the zone.</td>
</tr>
<tr>
<td>default_zone_role_type</td>
<td>The unique identifier for the user's default role when assigned to a zone.</td>
</tr>
<tr>
<td></td>
<td>- None: 00000000-0000-0000-0000-000000000000</td>
</tr>
<tr>
<td></td>
<td>- Zone Manager: 00000000-0000-0000-0000-000000000001</td>
</tr>
<tr>
<td></td>
<td>- User: 00000000-0000-0000-0000-000000000002</td>
</tr>
<tr>
<td>email</td>
<td>The user's email address.</td>
</tr>
<tr>
<td>first_name</td>
<td>The user's first name.</td>
</tr>
<tr>
<td>has_logged_in</td>
<td>True if the user has successfully logged in to the Console.</td>
</tr>
<tr>
<td>last_name</td>
<td>The user's last name.</td>
</tr>
<tr>
<td>page_number</td>
<td>The page number requested.</td>
</tr>
<tr>
<td>page_size</td>
<td>The page size requested.</td>
</tr>
<tr>
<td>role_type</td>
<td>The unique identifier defining the user's role in the Console.</td>
</tr>
<tr>
<td></td>
<td>- User: 00000000-0000-0000-0000-000000000001</td>
</tr>
<tr>
<td></td>
<td>- Administrator: 00000000-0000-0000-0000-000000000002</td>
</tr>
<tr>
<td></td>
<td>- Read-Only: 00000000-0000-0000-0000-000000000003</td>
</tr>
<tr>
<td></td>
<td>- Zone Manager: 00000000-0000-0000-0000-000000000004</td>
</tr>
<tr>
<td>tenant_id</td>
<td>The organization’s unique identifier for the Console.</td>
</tr>
<tr>
<td>total_number_of</td>
<td>The total number of resources.</td>
</tr>
</tbody>
</table>
### Field Name | Description
--- | ---
items | 
**total_pages** | The total number of pages that can be retrieved, based on the page size specified. 
**user_id** | The user's unique identifier for the Console. 
**zones** | The zones to which the user has access. This is an array of elements. 
- **id**: The unique identifier for the zone. 
- **role_type**: The user's role for this particular zone. 
  - None: 00000000-0000-0000-0000-000000000000 
  - Zone Manager: 00000000-0000-0000-0000-000000000001 
  - User: 00000000-0000-0000-0000-000000000002 
- **role_name**: The name of the user's role in this zone.

### Get User

Request information for a specific Console user resource belonging to a tenant.

**Service Endpoint:**

- /users/v2/{user_id | user_email_address}

Example with **user_id**: https://protectapi.cylance.com/users/v2/a2c0ac7a-a63d-4583-b646-ae10db9c9768

Example with a **user_email**: 
https://protectapi.cylance.com/users/v2/username@email.com

**Method:**

- HTTP/1.1 GET

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the user:read scope encoded

**Request:**

- None

**Response:**

200 OK
400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The user's unique identifier is not valid (when using a unique user ID)
- The user's email address specified is not a proper email address (when using the user's email address)

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The user resource cannot be found by the unique user ID or email address specified in the URL.

500 Internal Server Error - An unforeseeable error has occurred.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date_created</td>
<td>The date and time (in UTC) the Console user was created.</td>
</tr>
<tr>
<td>date_email_confirmed</td>
<td>The date and time (in UTC) when the user confirmed the email provided.</td>
</tr>
<tr>
<td>date_last_login</td>
<td>The date and time (in UTC) the user last logged in to the Console.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>date_modified</td>
<td>The date and time (in UTC) the Console user information was last updated.</td>
</tr>
<tr>
<td>default_zone_role_name</td>
<td>The name of the role for the user in the zone.</td>
</tr>
<tr>
<td>default_zone_role_type</td>
<td>The unique identifier for the user's default role when assigned to a zone.</td>
</tr>
</tbody>
</table>
|                                | - None: 00000000-0000-0000-0000-000000000000  
|                                | - Zone Manager: 00000000-0000-0000-0000-000000000001  
|                                | - User: 00000000-0000-0000-0000-000000000002                                                                                               |
| email                          | The user's email address.                                                                                                                        |
| first_name                     | The user's first name.                                                                                                                           |
| has_logged_in                  | True if the user has successfully logged in to the Console.                                                                                   |
| id                             | The user's unique identifier for the Console.                                                                                                  |
| last_name                      | The user's last name.                                                                                                                            |
| role_name                      | The name of the role.                                                                                                                            |
| role_type                      | The unique identifier defining the user's role in the Console.                                                                                 |
|                                | - User: 00000000-0000-0000-0000-000000000001  
|                                | - Administrator: 00000000-0000-0000-0000-000000000002  
|                                | - Read-Only: 00000000-0000-0000-0000-000000000003  
|                                | - Zone Manager: 00000000-0000-0000-0000-000000000004                                                                                           |
| tenant_id                      | The organization's unique identifier for the Console.                                                                                         |
| total_pages                    | The total number of pages that can be retrieved, based on the page size specified.                                                             |
| zones                          | The zones to which the user has access. This is an array of elements.                                                                          |
|                                | - id: The unique identifier for the zone.                                                                                                     |
|                                | - role_type: The user's role for this particular zone.                                                                                         |
|                                |   - None: 00000000-0000-0000-0000-000000000000  
|                                |   - Zone Manager: 00000000-0000-0000-0000-000000000001  
|                                |   - User: 00000000-0000-0000-0000-000000000002                                                                                                 |
|                                | - role_name: The name of the user's role in this zone.                                                                                         |

**Update User**

Update an existing Console user resource.

**Service Endpoint:**
Example with user id: https://protectapi.cylance.com/users/v2/a2c0ac7a-a63d-4583-b646-ae10db9c9768

Method:
- HTTP/1.1 PUT

Request Headers:
- Accept: application/json
- Content-Type: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the user:update scope encoded

Request:
```
{
  "email": "testuser@email.com",
  "user_role": "00000000-0000-0000-0000-000000000001",
  "first_name": "Test",
  "last_name": "User",
  "zones": [
    {
      "id": "d27ff5c4-5c0d-4f56-a00d-af1b297e440e",
      "role_type": "00000000-0000-0000-0000-000000000002"
    }
  ]
}
```

Response:

200 OK

400 Bad Request - Returned for the following reasons:
- The Tenant ID cannot be retrieved from the JWT token
- The user's unique identifier is not valid (when using a unique user ID)
- The user application role specified is not one of the accepted values
- The zones array is empty when the user application role is not Administrator

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The user resource cannot be found by the unique user ID or email address specified in the URL.

500 Internal Server Error - An unforeseeable error has occurred.
Request JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>email</td>
<td>The user's email address.</td>
</tr>
<tr>
<td>first_name</td>
<td>The user's first name.</td>
</tr>
<tr>
<td>last_name</td>
<td>The user's last name.</td>
</tr>
<tr>
<td>user_role</td>
<td>The unique identifier defining the user's role in the Console.</td>
</tr>
<tr>
<td></td>
<td>- User: 00000000-0000-0000-0000-000000000001</td>
</tr>
<tr>
<td></td>
<td>- Administrator: 00000000-0000-0000-0000-000000000002</td>
</tr>
<tr>
<td></td>
<td>- Read-Only: 00000000-0000-0000-0000-000000000003</td>
</tr>
<tr>
<td>zones</td>
<td>The zones to which the user has access. This is an array of elements.</td>
</tr>
<tr>
<td></td>
<td>- id: The unique identifier for the zone.</td>
</tr>
<tr>
<td></td>
<td>- role_type: The user's role for this particular zone.</td>
</tr>
<tr>
<td></td>
<td>- None: 00000000-0000-0000-0000-000000000000</td>
</tr>
<tr>
<td></td>
<td>- Zone Manager: 00000000-0000-0000-0000-000000000001</td>
</tr>
<tr>
<td></td>
<td>- User: 00000000-0000-0000-0000-000000000002</td>
</tr>
</tbody>
</table>

Note: If the user is an Administrator, the zones array is not required.

Note: To update a Zone Manager or change a user to a Zone Manager, set the user_role to User and set the Zones role_type to Zone Manager.

Delete User

Delete an existing Console user resource.

Service Endpoint:
- /users/v2/{user_id}

Example with user_id: https://protectapi.cylance.com/users/v2/a2c0ac7a-a63d-4583-b646-a610db9c9768

Method:
- HTTP/1.1 DELETE

Request Headers:
- Authorization: Bearer <JWT Token returned by Auth API> with the user:delete scope encoded

Request:
Send Invite Email

Request the Console login invitation email to be resent to a user who has not logged into the Console yet. The user must already be created, either using the Create User API or using the Console.

Service Endpoint:

- /users/v2/(user_email_address)/invite

Example:
https://protectapi.cylance.com/users/v2/username@email.com/invite

Method:

- HTTP/1.1 POST

Request Headers:

- Authorization: Bearer <JWT Token returned by Auth API> with the user:read scope encoded

Request:

None

Response:

200 OK - The email was successfully sent.

400 Bad Request - Returned for the following reasons:
- The Tenant ID cannot be retrieved from the JWT token
- The user's email address specified is not a proper email address
- The unique ID of the user triggering the invitation email to be sent cannot be retrieved from the JWT token
- The user to whom the invite email is to be sent has already logged in to the Console

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
404 Not Found - The user resource to send or resend the invitation email to is not found.
Send Request Password Email

Request for the Console reset password email to be sent or resent to an existing Console user.

Service Endpoint:

- /users/v2/(user_email_address)/resetpassword

Example:
https://protectapi.cylance.com/users/v2/username@email.com/resetpassword

Method:

- HTTP/1.1 POST

Request Headers:

- Authorization: Bearer <JWT Token returned by Auth API> with the user:read scope encoded

Request:

None

Response:

200 OK - The email was successfully sent.

400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The user's email address specified is not a proper email address
- The user to whom the reset password email is to be sent has not confirmed the email provided

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The user resource to send or resend the reset password email to is not found.

500 Internal Server Error - An unforeseeable error has occurred.
Device API
Get Devices

Request a page with a list of Console device resources belonging to a tenant, sorted by registration (created) date, in descending order (most recent device registered listed first). The page number and page size parameters are optional. When the values are not specified, these default to 1 and 10 respectively. The maximum page size that can be specified is 200 entries per page.

Note: When adding a new device to the Console, the new device will be seen at the top of the Device list when utilizing the Get Devices API call. When this occurs, all old devices will be moved down the list by one space. This causes the last device on the Device list to be move to the next page.

Potential Issue: Due to the above functionality, a race condition can occur where if a user is querying the Get Device API call for all devices in their Console and a new device is added to the Console during this time, the user may see "duplicate" devices appear in the API call. This occurs because of the above functionality, where a new device causes the last device on the Device list to be moved to the next page.

Example:

Device Name: TESTDEVICE is the last device on page 1 of the Device list page (using the Get Devices API call).

A new device is added to the Console.

If the Get Devices API call is querying all devices during this time frame, the returned query would show TESTDEVICE as being present on both page 1 and 2. This would appear to the user as if a "duplicate" device was present, but this is not the case and is simply due to the timing of the API call and the new device addition causing the same device to appear at the bottom of one page and at the top of the next.

A duplicate device would have the same device name, but a different device_ID. If a user spots a "duplicate" device in the Get Devices API call, it is recommended to compare the device_ID to confirm if it is truly a duplicate. If the two devices have the same device_ID, it is not an actual duplicate device respective to the Console.

For more information about Duplicate Devices, read this knowledge base article.

Recommendation: If a user is utilizing the Get Devices API call to iterate through all devices, it is suggested to query the first page of the device list again to catch any new devices that were potentially added during the iteration.
Service Endpoint:

- /devices/v2?page=m&page_size=n

Append the following optional query string parameters:

- page: The page number to request
- page_size: The number of device records to retrieve per page

For example, to return the first page with 100 devices:
https://protectapi.cylance.com/devices/v2?page=1&page_size=100

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the device:list scope encoded

Request:

None

Response:

200 OK

```json
{
  "page_number": "1",
  "page_size": "10",
  "total_pages": "1",
  "total_number_of_items": "1",
  "page_items": [
    {
      "id": "e378dacb-9324-453a-b8c6-5a8406952195",
      "name": "User-Laptop-A123",
      "state": "Offline",
      "agent_version": "2.0.1530",
      "policy": {
        "id": "d5c6d6a3-0599-4fb5-96bc-0fde7eacb6ea",
        "name": "Test Policy"
      },
      "date_first_registered": "2019-04-26T19:55:33",
      "ip_addresses": [
        "123.45.67.89"
      ],
      "mac_addresses": [
        "00-00-00-00-00-00"
      ]
    }
  ]
}
```
400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The page number or page size specified is less than or equal to zero, or the page size is greater than 200

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The device resources page requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent_version</td>
<td>The CylancePROTECT Agent version installed on the device.</td>
</tr>
<tr>
<td>date_first_registered</td>
<td>The date and time (in UTC) when the device record was created.</td>
</tr>
<tr>
<td>id</td>
<td>The endpoint's unique identifier.</td>
</tr>
<tr>
<td>ip_addresses</td>
<td>The list of IP addresses for the device.</td>
</tr>
<tr>
<td>mac_addresses</td>
<td>The list of MAC addresses for the device.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the device.</td>
</tr>
<tr>
<td>page_number</td>
<td>The page number requested.</td>
</tr>
<tr>
<td>page_size</td>
<td>The page size requested.</td>
</tr>
<tr>
<td>policy</td>
<td>The policy ID and name.</td>
</tr>
<tr>
<td>state</td>
<td>Signals whether the device is online or offline.</td>
</tr>
<tr>
<td>total_number_of_items</td>
<td>The total number of resources.</td>
</tr>
<tr>
<td>total_pages</td>
<td>The total number of pages that can be retrieved, based on the page size specified.</td>
</tr>
</tbody>
</table>
Get Device

Request a specific device resource belonging to a tenant.

Service Endpoint:

- `/devices/v2/{unique_device_id}`

Example: https://protectapi.cylance.com/devices/v2/e378dacb-9324-453a-b8c6-5a8406952195

Method:

- HTTP/1.1 GET

Request Headers:

- `Accept: application/json`
- `Authorization: Bearer <JWT Token returned by Auth API> with the device:read scope encoded`

Request:

None

Response:

200 OK

```json
{
    "id": "e378dacb-9324-453a-b8c6-5a8406952195",
    "name": "User-Laptop-A123",
    "host_name": "User-Laptop-A123",
    "os_version": "Microsoft Windows 10 Pro",
    "state": "Online",
    "agent_version": "2.0.1530",
    "policy": {
        "id": "d5c6d6a3-0599-4fb5-96bc-0f6c7eacb6ea",
        "name": "Test Policy"
    },
    "last_logged_in_user": "User-Laptop-A123\TestUser",
    "update_type": null,
    "update_available": false,
    "background_detection": false,
    "is_safe": true,
    "date_first_registered": "2019-04-04T16:45:04",
    "date_offline": "2019-04-04T16:45:04",
    "date_last_modified": "2019-04-04T16:45:04",
    "ip_addresses": ["123.45.67.89"],
    "mac_addresses": [
```
"00-00-00-00-00-00"
},
"distinguished_name": null
)

400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The device's unique identifier is not valid

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

500 Internal Server Error - An unforeseeable error has occurred.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent_version</td>
<td>The CylancePROTECT Agent version installed on the device.</td>
</tr>
<tr>
<td>background_detection</td>
<td>If true, the Agent is currently running a Background Threat Detection scan.</td>
</tr>
<tr>
<td>date_first_registered</td>
<td>The date and time (in UTC) when the device record was created.</td>
</tr>
<tr>
<td>date_last_modified</td>
<td>The date and time (in UTC) when the device record was last modified.</td>
</tr>
<tr>
<td>date_offline</td>
<td>The date and time (in UTC) when the device last communicated with the Console.</td>
</tr>
<tr>
<td>distinguished_name</td>
<td>The unique identifier for the device in the Lightweight Directory Access Protocol (LDAP).</td>
</tr>
<tr>
<td>host_name</td>
<td>The hostname for the device.</td>
</tr>
<tr>
<td>id</td>
<td>The unique identifier for the device.</td>
</tr>
<tr>
<td>ip_addresses</td>
<td>The list of IP addresses for the device.</td>
</tr>
<tr>
<td>is_safe</td>
<td>If true, there are no outstanding threats.</td>
</tr>
<tr>
<td>last_logged_in_user</td>
<td>The ID of the user who logged in last on to the device.</td>
</tr>
<tr>
<td>mac_addresses</td>
<td>The list of MAC addresses for the device.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the device.</td>
</tr>
<tr>
<td>os_version</td>
<td>The operating system and version.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>policy</td>
<td>The name of the policy assigned to the device.</td>
</tr>
<tr>
<td>state</td>
<td>The device is online or offline.</td>
</tr>
<tr>
<td>update_available</td>
<td>If true, an Agent update is available for the device based on the update type (Phase).</td>
</tr>
<tr>
<td>update_type</td>
<td>The update phase on which the device is scheduled.</td>
</tr>
</tbody>
</table>

**Get Device by MAC Address**

Request a specific device resource belonging to a tenant by using the MAC address of the device.

**Service Endpoint:**

- `/devices/v2/macaddress/{mac_address}

**Example:** https://protectapi.cylance.com/devices/v2/macaddress/28-F1-0E-45-AB-54

**Example:**

**Method:**

- HTTP/1.1 GET

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the device:read scope encoded

**Request:**

None

**Response:**

200 OK

```
{
  "id": "e378dacb-9324-453a-b8c6-5a8406952195",
  "name": "User-Laptop-A123",
  "host_name": "User-Laptop-A123",
  "os_version": "Microsoft Windows 10 Pro",
  "state": "Online",
  "agent_version": "2.0.1530",
```
"policy": {
    "id": "d5c6d6a3-0599-4fb5-96bc-0fdc7eacb6ea",
    "name": "Test Policy",
    "last_logged_in_user": "User-Laptop-A123\TestUser",
    "update_type": null,
    "update_available": false,
    "background_detection": false,
    "is_safe": true,
    "date_first_registered": "2019-04-04T16:45:04",
    "date_offline": "2019-04-04T16:45:04",
    "date_last_modified": "2019-04-04T16:45:04",
    "ip_addresses": [
        "123.45.67.89"
    ],
    "mac_addresses": [
        "00-00-00-00-00"
    ],
    "distinguished_name": null
}

400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The device's unique identifier is not valid

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

500 Internal Server Error - An unforeseeable error has occurred.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent_version</td>
<td>The CylancePROTECT Agent version installed on the device.</td>
</tr>
<tr>
<td>background_detection</td>
<td>If true, the Agent is currently running a Background Threat Detection scan.</td>
</tr>
<tr>
<td>date_first_registered</td>
<td>The date and time (in UTC) when the device record was created.</td>
</tr>
<tr>
<td>date_last_modified</td>
<td>The date and time (in UTC) when the device record was last modified.</td>
</tr>
<tr>
<td>date_offline</td>
<td>The date and time (in UTC) when the device last communicated with the Console.</td>
</tr>
<tr>
<td>distinguished_name</td>
<td>The unique identifier for the device in the Lightweight Directory Access Protocol (LDAP).</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>host_name</td>
<td>The hostname for the device.</td>
</tr>
<tr>
<td>id</td>
<td>The unique identifier for the device.</td>
</tr>
<tr>
<td>ip_addresses</td>
<td>The list of IP addresses for the device.</td>
</tr>
<tr>
<td>is_safe</td>
<td>If true, there are no outstanding threats.</td>
</tr>
<tr>
<td>last_logged_in_user</td>
<td>The ID of the user who logged in last on to the device.</td>
</tr>
<tr>
<td>mac_addresses</td>
<td>The list of MAC addresses for the device.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the device.</td>
</tr>
<tr>
<td>os_version</td>
<td>The operating system and version.</td>
</tr>
<tr>
<td>policy</td>
<td>The name of the policy assigned to the device.</td>
</tr>
<tr>
<td>state</td>
<td>The device is online or offline.</td>
</tr>
<tr>
<td>update_available</td>
<td>If true, an Agent update is available for the device based on the update type (Phase).</td>
</tr>
<tr>
<td>update_type</td>
<td>The update phase on which the device is scheduled.</td>
</tr>
</tbody>
</table>

**Get Device by Hostname**

Request a specific device resource belonging to a tenant by using the hostname of the device (DNS name).

**Note:** The hostname ("host_name") may not be the same as the name ("name") displayed by Cylance. The hostname is created by the operating system, while the name can be changed in the Cylance Console or API.

**Service Endpoint:**

- `/devices/v2/hostname/{host_name}`


**Method:**

- HTTP/1.1 GET

**Request Headers:**
Accept: application/json
Authorization: Bearer <JWT Token returned by Auth API> with the device:read scope encoded

Request:
None

Response:
200 OK

```
{
    "id": "e378dacb-9324-453a-b8c6-5a8406952195",
    "name": "Executive Laptop",
    "host_name": "User-Laptop-A123",
    "os_version": "Microsoft Windows 10 Pro",
    "state": "Online",
    "agent_version": "2.0.1530",
    "policy": {
        "id": "d5c6d6a3-0599-4fb5-96bc-0f8c7eacb6ea",
        "name": "Test Policy"
    },
    "last_logged_in_user": "User-Laptop-A123\TestUser",
    "update_type": null,
    "update_available": false,
    "background_detection": false,
    "is_safe": true,
    "date_first_registered": "2019-04-04T16:45:04",
    "date_offline": "2019-04-04T16:45:04",
    "date_last_modified": "2019-04-04T16:45:04",
    "ip_addresses": ["123.45.67.89"],
    "mac_addresses": ["00-00-00-00-00-00"],
    "distinguished_name": null
}
```

400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The device's unique identifier is not valid

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
500 Internal Server Error - An unforeseeable error has occurred.

Response JSON Schema Descriptions
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent_version</td>
<td>The CylancePROTECT Agent version installed on the device.</td>
</tr>
<tr>
<td>background_detection</td>
<td>If true, the Agent is currently running a Background Threat Detection scan.</td>
</tr>
<tr>
<td>date_first_registered</td>
<td>The date and time (in UTC) when the device record was created.</td>
</tr>
<tr>
<td>date_last_modified</td>
<td>The date and time (in UTC) when the device record was last modified.</td>
</tr>
<tr>
<td>date_offline</td>
<td>The date and time (in UTC) when the device last communicated with the Console.</td>
</tr>
<tr>
<td>distinguished_name</td>
<td>The unique identifier for the device in the Lightweight Directory Access Protocol (LDAP).</td>
</tr>
<tr>
<td>host_name</td>
<td>The hostname for the device.</td>
</tr>
<tr>
<td>id</td>
<td>The unique identifier for the device.</td>
</tr>
<tr>
<td>ip_addresses</td>
<td>The list of IP addresses for the device.</td>
</tr>
<tr>
<td>is_safe</td>
<td>If true, there are no outstanding threats.</td>
</tr>
<tr>
<td>last_logged_in_user</td>
<td>The ID of the user who logged in last on to the device.</td>
</tr>
<tr>
<td>mac_addresses</td>
<td>The list of MAC addresses for the device.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the device.</td>
</tr>
<tr>
<td>os_version</td>
<td>The operating system and version.</td>
</tr>
<tr>
<td>policy</td>
<td>The name of the policy assigned to the device.</td>
</tr>
<tr>
<td>state</td>
<td>The device is online or offline.</td>
</tr>
<tr>
<td>update_available</td>
<td>If true, an Agent update is available for the device based on the update type (Phase).</td>
</tr>
<tr>
<td>update_type</td>
<td>The update phase on which the device is scheduled.</td>
</tr>
</tbody>
</table>

**Update Device**

Update a specific device resource belonging to a tenant.

**Service Endpoint:**

- `/devices/v2/{unique_device_id}`
Example: https://protectapi.cylance.com/devices/v2/e378dacb-9324-453ab8c6-5a8406952195

Method:
- HTTP/1.1 PUT

Request Headers:
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the device:update scope encoded
- Content-Type: application/json

Request:
```
{
    "name": "User-Laptop-A123",
    "policy_id": "d5c6d6a3-0599-4fb5-96bc-0fdec7eacb6ea",
    "add_zone_ids": ["d27ff5c4-5c0d-4f56-a00d-a1fb297e440e"],
    "remove_zone_ids": ["639db7f7-c7f9-488d-b834-41c4522b32b6"
}
```

Response:
200 OK
400 Bad Request - Returned for the following reasons:
- The Tenant ID cannot be retrieved from the JWT token
- The device's unique identifier is not valid

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
404 Not Found - The device resource to update doesn't exist.
500 Internal Server Error - An unforeseeable error has occurred.

Request JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>add_zone_ids</td>
<td>The list of zone identifiers which the device is to be assigned.</td>
</tr>
</tbody>
</table>
## Get Device Threat

Request a page with a list of threats found on a specific device. The page number and page size parameters are optional. When the values are not specified, these default to 1 and 10 respectively. The maximum page size that can be specified is 200 entries per page.

**Service Endpoint:**

```
/devices/v2/(unique_device_id)/threats?page=m&page_size=n
```

Append the following optional query string parameters:

- `page`: The page number to request
- `page_size`: The number of device records to retrieve per page

**Example:** [https://protectapi.cylance.com/devices/v2/e378dacb-9324-453a-b8c6-5a8406952195/threats?page=1&page_size=100](https://protectapi.cylance.com/devices/v2/e378dacb-9324-453a-b8c6-5a8406952195/threats?page=1&page_size=100)

**Method:**

- HTTP/1.1 GET

**Request Headers:**

- `Accept`: application/json
- `Authorization: Bearer <JWT Token returned by Auth API>` with the device:threatlist scope encoded

**Request:**

None

**Response:**

200 OK

```json
{
  "page_number": "1",
```
"page_size": "10",
"total_pages": "1",
"total_number_of_items": "1",
"page_items": [
{
    "name": "threat.exe",
    "sha256": "bf17366ee3bb8068a9ad70fc9e68496e7e311a055bf4ffeef53cc5d29ccce52",
    "file_status": "Suspicious",
    "file_path": "C:\Some\File\Path\threat.exe",
    "cylance_score": "-0.900",
    "classification": "Malware",
    "sub_classification": "Backdoor",
    "date_found": "2019-05-08T20:55:27"
}
]

400 Bad Request - Returned for the following reasons:
- The Tenant ID cannot be retrieved from the JWT token
- The device's unique identifier is not valid
- The page number or page size specified is less than or equal to zero, or the page size is greater than 200

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The identifier specified doesn't belong to a device resource in the system.

500 Internal Server Error - An unforeseeable error has occurred.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>classification</td>
<td>The threat classification assigned by Cylance.</td>
</tr>
<tr>
<td>cylance_score</td>
<td>The Cylance score assigned to the threat. The Cylance API returns a raw score of -1 to 1. Threats have a negative raw score, while safe files have a positive raw score. The Cylance Console only displays threats and uses a score of 1 to 100. A raw score of -1 equals a Console score of 100.</td>
</tr>
<tr>
<td>date_found</td>
<td>The date and time (in UTC) when the threat was found on the device.</td>
</tr>
<tr>
<td>file_path</td>
<td>The file path to the threat, includes the file name.</td>
</tr>
<tr>
<td>file_status</td>
<td>The current status of the file on the device. This can be one of the following:</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>n</td>
<td>Default (0) (Unsafe)</td>
</tr>
<tr>
<td></td>
<td>Quarantined (1)</td>
</tr>
<tr>
<td></td>
<td>Whitelisted (2)</td>
</tr>
<tr>
<td></td>
<td>Suspicious (3) (Abnormal)</td>
</tr>
<tr>
<td></td>
<td>File Removed (4) (Delete) - The file was removed from the Console.</td>
</tr>
<tr>
<td></td>
<td>Corrupt (5) - The file could not be scanned. The file could be corrupt or malformed.</td>
</tr>
</tbody>
</table>

- **name**: The name of the threat.
- **page_number**: The page number requested.
- **page_size**: The page size requested.
- **sha256**: The SHA256 hash for the threat.
- **sub_classification**: The threat sub-classification assigned by Cylance.
- **total_pages**: The total number of pages that can be retrieved, based on the page size specified.
- **total_number_of_items**: The total number of resources.

**Update Device Threat**

Update the status (waive or quarantine) of a convicted threat.

**Note**: To update a threat on a device requires the Modify permission for the Threats privilege in an Integration. See Authorization below.

**Service Endpoint**:

- `/devices/v2/{unique_device_id}/threats`

**Example**: `https://protectapi.cylance.com/devices/v2/e378dacb-9324-453a-b8c6-5a8406952195/threats`

**Method**:

- HTTP/1.1 POST

**Request Headers**: 
Accept: application/json

Authorization: Bearer <JWT Token returned by Auth API> with the threat:update scope encoded

Content-Type: application/json

Request:

```json
{
    "threat_id": "bf17366ee3bb8068a9ad70fc9e68496e7e311a055bf4ffeef53cc5d29ccce52",
    "event": "Quarantine"
}
```

Response:

200 OK

400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The device's unique identifier is not valid

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The device resource to update doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>The requested status update for the convicted threat.</td>
</tr>
<tr>
<td></td>
<td>- Quarantine</td>
</tr>
<tr>
<td></td>
<td>- Waive</td>
</tr>
<tr>
<td>threat_id</td>
<td>The SHA256 hash of the convicted threat.</td>
</tr>
</tbody>
</table>

Get Zone Devices

Request a page with a list of Console device resources belonging to a zone, sorted by registration (created) date, in descending order (most recent registered listed first). The page number and page size parameters are optional. When the values are not specified, these
default to 1 and 10 respectively. The maximum page size that can be specified is 200 entries per page.

**Service Endpoint:**

- /devices/v2/{unique_zone_id}/devices?page=m&page_size=n

Append the following optional query string parameters:

- page: The page number to request
- page_size: The number of device records to retrieve per page

**Example:** https://protectapi.cylance.com/devices/v2/d27ff5c4-5c0d-4f56-a00d-alfb297e440e/devices?page=1&page_size=100

**Method:**

- HTTP/1.1 GET

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the device:list scope encoded

**Request:**

None

**Response:**

200 OK

```
{
    "page_number": "1",
    "page_size": "10",
    "total_pages": "1",
    "total_number_of_items": "1",
    "page_items": [
        {
            "id": "e378dabc-9324-453a-b8c6-5a8406952195",
            "name": "User-Laptop-A123",
            "policy_id": "d5c6d6a3-0599-4fb5-96bc-0f0e7eac6ea"
        }
    ]
}
```

400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The device's unique identifier is not valid
- The page number or page size specified is less than or equal to zero, or the page size is greater than 200

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The zone resource requested is not a valid ID.

500 Internal Server Error - An unforeseeable error has occurred.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The unique identifier for the device.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the device.</td>
</tr>
<tr>
<td>page_number</td>
<td>The page number requested.</td>
</tr>
<tr>
<td>page_size</td>
<td>The page size requested.</td>
</tr>
<tr>
<td>policy_id</td>
<td>The unique identifier for the policy to which the policy is currently assigned. This can be null.</td>
</tr>
<tr>
<td>total_number_of_items</td>
<td>The total number of resources.</td>
</tr>
<tr>
<td>total_pages</td>
<td>The total number of pages that can be retrieved, based on the page size specified</td>
</tr>
</tbody>
</table>

**Get Agent Installer Link**

Request a secured link to download the Agent installer.

**Service Endpoint:**

```
/devices/v2/installer?product=p&os=o&package=k&architecture=a&build=v
```

Example:


**Method:**

- HTTP/1.1 GET

**Request Headers:**
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the device:read scope encoded

**Request:**

Append the following optional query string parameters:

- **product** (required for all operating systems): Specify the Cylance product installer to download. The allowed values are:
  - Protect
  - Optics
  - Protect+Optics

  **Note:** To download the CylancePROTECT+CylanceOPTICS installer, use the following example (Windows only). This downloads the latest CylancePROTECT+CylanceOPTICS installer available to the tenant. Other versions of the CylancePROTECT+CylanceOPTICS installer must be downloaded from the Console.

  EXE Example:

  32-bit MSI Example:

  64-bit MSI Example:

- **os** (required for all operating systems): Specify the operating system (OS) family. The allowed values are:
  - AmazonLinux1
  - AmazonLinux2
  - CentOS7
  - Linux

  **Note:** Use Linux as the OS family for CentOS6.
  - Mac
  - Suse11
- Suse12
- Ubuntu1404
- Ubuntu1604
- Ubuntu1804
- Windows

- architecture (required for Linux and Windows): Specify the target architecture. The allowed values are:
  - X86
  - X64
  - AmazonLinux1
  - AmazonLinux2
  - CentOS6
  - CentOS6UI
  - CentOS7
  - CentOS7UI
  - Suse11
  - Suse11UI
  - Suse12
  - Suse12UI
  - Ubuntu1404
  - Ubuntu1404UI
  - Ubuntu1604
  - Ubuntu1604UI
  - Ubuntu1804
  - Ubuntu1804UI

- package (required for macOS and Windows): Specify the installer format. The allowed values are:
  - Exe (Windows only)
  - Msi (Windows only)
- Dmg (macOS only)
- Pkg (macOS only)


Response:

200 OK

```json
{
    "url": "https://cylance/download/url"
}
```

400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The requested payload failed validation

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The requested link resource does not exist.

500 Internal Server Error - An unforeseeable error has occurred.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>The URL you can use to download the requested Agent installer. The API call only provides the URL, it does not download the installer for you.</td>
</tr>
</tbody>
</table>

**Delete Devices**

Delete one or more devices from a tenant.

**Note:** This is an asynchronous operation and could take up to two hours to delete the devices.

If a callback URL is provided, the callback will occur when deletion is complete.

**Service Endpoint:**

- /devices/v2

Example: https://protectapi.cylance.com/devices/v2

**Method:**
HTTP/1.1 DELETE

**Note:** For clients who do not support DELETE, see the note below.

**Request Header:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the device:delete scope encoded.
- Content-Type: application/json

**Request:**

```
{
    "device_ids": [
        "e378dacb-9324-453a-b8c6-5a8406952195"
    ],
    "callback_url": ""
}
```

**Response:**

202 Accepted

```
{
    "request_id": ""
}
```

400 Bad Request - Returned for the following reasons:

- The tenant ID could not be retrieved from the JWT token
- The device IDs are not valid

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The devices do not exist.

413 Payload Too Large - The number of resources specified in the request is too large.

500 Internal Server Error - An unforeseeable error has occurred.

**Request JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>callback_url</td>
<td>The URL of the callback upon completion. This is optional.</td>
</tr>
</tbody>
</table>
### Field Name | Description
--- | ---
device_ids | The unique identifiers for the devices to be deleted.  
- All device IDs should be well formed GUIDs. Non-conforming values will be removed from the request.  
- The maximum number of device IDs per request is 20.

### Response JSON Schema Descriptions

| Field Name | Description |
--- | ---
request_id | The unique identifier for the deletion request.

**Note:** Not all clients support sending a DELETE request. For this instance, use the following POST instead.

- **Service Endpoint:** /devices/v2/delete  
  Example: https://protectapi.cylance.com/devices/v2/delete

- **Method:** HTTP/1.1 POST

Delete this text and replace it with your own content.
Global List API
Get Global List

Request a page with a list of global list resources for a tenant, sorted by the date when the hash was added to the global list, in descending order (most recent policy listed first). The page number and page size parameters are optional. When the values are not specified, these default to 1 and 10 respectively. The maximum page size that can be specified is 200 entries per page. The listTypeId parameter is required and can be 0 (Global Quarantined) or 1 (Global Safe).

Service Endpoint:

- /globallists/v2?listTypeId={0|1}&page-m&page_size=n

Append the following optional query string parameters:

- listTypeId: The type of list to retrieve.
  - 0 = Global Quarantine list
  - 1 = Global Safe list
- page: The page number to request
- page_size: The number of device records to retrieve per page

For example, to return the first page with up to 100 list items:
https://protectapi.cylance.com/globallists/v2?listTypeId=0&page=1&page_size=100

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the globallist:list scope encoded

Request:

None

Response:

200 OK

```json
{
  "page_number": "1",
  "page_size": "10",
  "total_pages": "1",
}
```
"total_number_of_items": "1",
"page_items": [
  {
    "name": "threat.exe",
    "sha256": "bf1736ee3bb8068a9ad70fc9e68496e7e311a055bf4ffeef53cc5d29ccce52",
    "md5": "d41d8cd98f00b204e9800988ecf8427e",
    "cylance_score": "-1",
    "av_industry": null,
    "classification": "Malware",
    "sub_classification": "Trojan",
    "list_type": "GlobalQuarantine",
    "category": "None",
    "added": "2019-05-07T16:41:5",
    "added_by": "a2c0ac7a-a63d-4583-b646-ae10db9c9768",
    "reason": "Test"
  }
]

400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The page number or page size specified is less than or equal to zero

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The global list page requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>added</td>
<td>The timestamp when the file was added to the list.</td>
</tr>
<tr>
<td>added_by</td>
<td>The tenant user ID who added the file to the list.</td>
</tr>
<tr>
<td>av_industry</td>
<td>The score provided by the antivirus industry.</td>
</tr>
<tr>
<td>category</td>
<td>The category for the list specified (for the Global Safe list only).</td>
</tr>
<tr>
<td></td>
<td>- Admin Tool</td>
</tr>
<tr>
<td></td>
<td>- Commercial Software</td>
</tr>
<tr>
<td></td>
<td>- Drivers</td>
</tr>
<tr>
<td></td>
<td>- Internal Application</td>
</tr>
<tr>
<td></td>
<td>- None</td>
</tr>
<tr>
<td></td>
<td>- Operating System</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>classification</td>
<td>The threat classification assigned by Cylance.</td>
</tr>
<tr>
<td>cylance_score</td>
<td>The Cylance score assigned to the threat. The Cylance API returns a raw score of -1 to 1. Threats have a negative raw score, while safe files have a positive raw score. The Cylance Console only displays threats and uses a score of 1 to 100. A raw score of -1 equals a Console score of 100.</td>
</tr>
<tr>
<td>list_type</td>
<td>The list type to which the threat belongs (Global Quarantine or Global Safe).</td>
</tr>
<tr>
<td>md5</td>
<td>The MD5 hash for the threat.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the threat.</td>
</tr>
<tr>
<td>page_number</td>
<td>The page number requested.</td>
</tr>
<tr>
<td>page_size</td>
<td>The page size requested.</td>
</tr>
<tr>
<td>reason</td>
<td>The reason why the file was added to the list.</td>
</tr>
<tr>
<td>sha256</td>
<td>The SHA256 hash for the threat.</td>
</tr>
<tr>
<td>sub_classification</td>
<td>The threat sub-classification assigned by Cylance.</td>
</tr>
<tr>
<td>total_pages</td>
<td>The total number of pages that can be retrieved, based on the page size specified.</td>
</tr>
<tr>
<td>total_number_of_items</td>
<td>The total number of resources.</td>
</tr>
</tbody>
</table>

**Add to Global List**

Add a convicted threat to either the Global Quarantine or the Global safe list for a particular tenant.

**Service Endpoint:**

- /globallists/v2

**Example:** https://protectapi.cylance.com/globallists/v2
Method:
- HTTP/1.1 POST

Request Headers:
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the globalist:create scope encoded

Request:

```json
{
"sha256": "bf17366ee3bb8068a9ad70fc9e68496e7e311a055bf4ffeef53cc5d29ccce52",
"list_type": "GlobalSafe",
"category": "CommercialSoftware",
"reason": "Test"
}
```

Response:

200 OK

400 Bad Request - Returned for the following reasons:
- The Tenant ID cannot be retrieved from the JWT token
- The requested payload failed validation

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

409 Conflict - The threat specified already exists in the intended list.

500 Internal Server Error - An unforeseeable error has occurred.

Request JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>category</td>
<td>This field is required only if the list_type value is GlobalSafe. The value can be one of the following:</td>
</tr>
<tr>
<td></td>
<td>AdminTool</td>
</tr>
<tr>
<td></td>
<td>CommercialSoftware</td>
</tr>
<tr>
<td></td>
<td>Drivers</td>
</tr>
<tr>
<td></td>
<td>InternalApplication</td>
</tr>
<tr>
<td></td>
<td>OperatingSystem</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SecuritySoftware</td>
<td>There are no spaces in the Category name when adding to a Global List.</td>
</tr>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

**list_type**
The list type to which the threat belongs (GlobalQuarantine or GlobalSafe).

**reason**
The reason why the file was added to the list.

**sha256**
The SHA256 hash for the threat.

---

**Delete From Global List**

Remove a convicted threat from either the Global Quarantine or the Global Safe list for a particular tenant.

**Service Endpoint:**

- /globallists/v2

**Example:** https://protectapi.cylance.com/globallists/v2

**Method:**

- HTTP/1.1 DELETE

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the globallist:delete scope encoded

**Request:**

```
{
  "sha256": "bf17366ee3bb8068a9ad70fc9e68496e7e311a055bf4ffeef53cc5d29ccce52",
  "list_type": "GlobalSafe"
}
```

**Response:**

200 OK

400 Bad Request - Returned for the following reasons:
- The Tenant ID cannot be retrieved from the JWT token
- The provided SHA256 is invalid
- The provided list type is not supported

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
404 Not Found - The threat specified does not exist in the intended list.
500 Internal Server Error - An unforeseeable error has occurred.

**Request JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list_type</td>
<td>The list type to which the threat belongs (GlobalQuarantine or GlobalSafe).</td>
</tr>
<tr>
<td>sha256</td>
<td>The SHA256 hash for the threat.</td>
</tr>
</tbody>
</table>
Policy API
Get Policies

Request a page with a list of Console policies belonging to a tenant, sorted by modified date, in descending order (most recently modified policy listed first). The page number and page size parameters are optional. When the values are not specified, these default to 1 and 10 respectively.

Note: When a policy is created, the modified date is the same as the created date, until the policy is modified.

Service Endpoint:

- /policies/v2?page=m&page_size=n

Append the following optional query string parameters:

- page: The page number to request
- page_size: The number of device records to retrieve per page

For example, to return the first page with up to 100 policies:
https://protectapi.cylance.com/policies/v2?page=1&page_size=100

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the policy:list scope encoded

Request:

None

Response:

200 OK

```json
{
    "page_number": "1",
    "page_size": "10",
    "total_pages": "1",
    "total_number_of_items": "1",
    "page_items": [
        {
            "id": "d5c6d6a3-0599-4fb5-96bc-0fcd7eacb6ea",
            "name": "Test Policy",
            "device_count": "1",
        }
    ]
}
```
"zone_count": "1",
"date_added": "2019-05-08T21:32:46.9860259",
"date_modified": "2019-05-08T21:32:46.9860259"
}
]

400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The page number or page size specified is less than or equal to zero, or the page size is greater than 100

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The device resources page requested doesn’t exist.

500 Internal Server Error - An unforeseeable error has occurred.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>date_added</td>
<td>The date and time (in UTC) when the Console policy resource was first created.</td>
</tr>
<tr>
<td>date_modified</td>
<td>The date and time (in UTC) when the Console policy resource was last modified.</td>
</tr>
<tr>
<td>device_count</td>
<td>The number of devices assigned to this policy.</td>
</tr>
<tr>
<td>id</td>
<td>The unique ID for the policy resource.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the policy.</td>
</tr>
<tr>
<td>page_number</td>
<td>The page number requested.</td>
</tr>
<tr>
<td>page_size</td>
<td>The page size requested.</td>
</tr>
<tr>
<td>total_number_of_items</td>
<td>The total number of resources.</td>
</tr>
<tr>
<td>total_pages</td>
<td>The total number of pages that can be retrieved based on the page size specified.</td>
</tr>
<tr>
<td>zone_count</td>
<td>The number of zones assigned to this policy.</td>
</tr>
</tbody>
</table>
Get Policy

Get details for a policy, using the policy ID.

Service Endpoint:

- /policies/v2/{policy_id}

Example: https://protectapi.cylance.com/policies/v2/d5c6d6a3-0599-4fb5-96bc-0fdc7eacb6ea

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the policy:read scope encoded

Request:

None

Response:

200 OK

{  
  "policy": [  
    {  
      "value": "1",
      "name": "auto_blocking"
    },  
    {  
      "value": "1",
      "name": "auto_uploading"
    },  
    {  
      "value": "500",
      "name": "threat_report_limit"
    },  
    {  
      "name": "low_confidence_threshold",
      "value": "-600"
    },  
    {  
      "value": "2",
      "name": "full_disc_scan"
    }
  ]
}
"name": "watch_for_new_files",
"value": "1"
},
{
 "value": "1",
 "name": "memory_exploit_detection"
},
{
 "value": "1",
 "name": "trust_files_in_scan_exception_list"
},
{
 "name": "logpolicy",
 "value": "0"
},
{
 "name": "script_control",
 "value": "1"
},
{
 "value": "1",
 "name": "prevent_service_shutdown"
},
{
 "name": "scan_max_archive_size",
 "value": "0"
},
{
 "name": "sample_copy_path",
 "value": "\\\server_name\\shared_folder"
},
{
 "name": "kill_running_threats",
 "value": "1"
},
{
 "name": "show_notifications",
 "value": "1"
},
{
 "value": "1000",
 "name": "optics_set_disk_usage_maximum_fixed"
},
{
 "name": "optics_malware_auto_upload",
 "value": "1"
},
{
 "value": "1",
 "name": "optics_memory_defense_auto_upload"
}
{
  "value": "0",
  "name": "optics_script_control_auto_upload"
},
{
  "value": "0",
  "name": "optics_application_control_auto_upload"
},
{
  "name": "device_control",
  "value": "1"
},
{
  "name": "optics_sensors_dns_visibility",
  "value": "0"
},
{
  "value": "0",
  "name": "optics_sensors_private_network_address_visibility"
},
{
  "name": "optics_sensors_windows_event_log_visibility",
  "value": "0"
},
{
  "name": "optics_sensors_advanced_powershell_visibility",
  "value": "0"
},
{
  "name": "optics_sensors_advanced_wmi_visibility",
  "value": "0"
},
{
  "name": "optics_sensors_advanced_executable_parsing",
  "value": "0"
},
{
  "name": "optics_sensors_enhanced_process_hooking_visibility",
  "value": "0"
},
{
  "value": "1",
  "name": "optics"
},
{
  "name": "auto_delete",
  "value": "1"
},
{
  "name": "days_until_deleted",
  "value": "14"
}
"name": "pdf_auto_uploading",
"value": "0"
},
{
"name": "ole_auto_uploading",
"value": "0"
},
{
"name": "docx_auto_uploading",
"value": "0"
},
{
"value": "0",
"name": "python_auto_uploading"
},
{
"value": "0",
"name": "autoit_auto_uploading"
},
{
"name": "powershell_auto_uploading",
"value": "0"
},
{
"name": "custom_thumbprint",
"value": null
},
{
"value": [
   "c:\temp"
],
"name": "scan_exception_list"
},
{
"name": "optics_show_notifications",
"value": "1"
}
],
"device_control": {
"configurations": [

{"device_class": "AndroidUSB",
 "control_mode": "FullAccess"
 },
{"control_mode": "FullAccess",
 "device_class": "iOS"
},
{"control_mode": "FullAccess",
 "device_class": "StillImage"
],


```json
{
    "device_class": "USBCDDVDRW",
    "control_mode": "FullAccess"
},
{
    "control_mode": "FullAccess",
    "device_class": "USBDrive"
},
{
    "device_class": "VMWareMount",
    "control_mode": "FullAccess"
},
{
    "control_mode": "FullAccess",
    "device_class": "WPD"
}
],
"exclusion_list": [
{
    "vendor_id": "1234",
    "comment": "Test device control exclusion",
    "serial_number": "987654321",
    "product_id": "5678",
    "control_mode": "FullAccess",
    "date_added": "2019-03-07T17:58:36.484Z"
}
]
},
"memoryviolation_actions": {
    "memory_violations": [
    {
        "violation_type": "lsassread",
        "action": "Terminate"
    },
    {
        "violation_type": "outofprocessunmapmemory",
        "action": "Terminate"
    },
    {
        "violation_type": "stackpivot",
        "action": "Terminate"
    },
    {
        "violation_type": "stackprotect",
        "action": "Terminate"
    },
    {
        "violation_type": "outofprocessoverwritecode",
        "action": "Terminate"
    },
    {
        "action": "Terminate",
        "violation_type": "outofprocesscreatethread"
    }
}

{
  "violation_type": "overwritecode",
  "action": "Terminate"
},
{
  "action": "Terminate",
  "violation_type": "outofprocesswritepe"
},
{
  "violation_type": "outofprocessallocation",
  "action": "Terminate"
},
{
  "violation_type": "outofprocessmap",
  "action": "Terminate"
},
{
  "violation_type": "outofprocesswrite",
  "action": "Terminate"
},
{
  "action": "Terminate",
  "violation_type": "outofprocessapc"
}]
"memory_violations_ext": [
  {
    "violation_type": "dyldinjection",
    "action": "Terminate"
  },
  {
    "violation_type": "trackdataread",
    "action": "Terminate"
  },
  {
    "action": "Terminate",
    "violation_type": "zeroallocate"
  },
  {
    "action": "Terminate",
    "violation_type": "maliciouspayload"
  }
],
"memory_exclusion_list": [
  "\temp"
],
"script_control": {
  "powershell_settings": {
    "control_mode": "Block",
    "console_mode": "Block"
  }
}
"global_settings": {
  "control_mode": "Alert",
  "allowed_folders": [
    "\temp_scriptcontrol"
  ],
",  "activescript_settings": {
    "control_mode": "Alert"
  },
"filetype_actions": {
  "suspicious_files": [
    { "actions": "3",
      "file_type": "executable"
    }
  ],
  "threat_files": [
    { "actions": "3",
      "file_type": "executable"
    }
  ]
},
"logpolicy": {
  "retentiondays": "30",
  "log_upload": null,
  "maxlogsize": "100"
},
"checksum": "",
"file_exclusions": [],
"policy_name": "Test Policy",
"appcontrol": {
  "lockdown": [
    { "action": "deny",
      "lockdown_type": "executionfromexternaldrives"
    },
    { "lockdown_type": "pechange",
      "action": "deny"
    }
  ],
  "allowed_folders": [
    "c:\temp_appcontrol"
  ],
  "changewindow_enabled": "1"
},
"policy_id": "d5c6d6a3-0599-4fb5-96bc-0fdc7eacb6ea",
"policy_utctimestamp": "/Date(1557768200200+0000)="/}
### Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>appcontrol</td>
<td>Application Control allows restricting any changes to applications on a device. Only the applications that exist on a device before enabling Application Control are allowed to execute on that device. Any new applications, as well as changes to the executables of existing applications, will be denied. The Agent Updater will also be disabled when Application Control is enabled.</td>
</tr>
<tr>
<td>checksum</td>
<td>Used for detecting errors that may have occurred during transmission or storage of data.</td>
</tr>
<tr>
<td>device_control</td>
<td>Device Control</td>
</tr>
<tr>
<td>control_mode</td>
<td>Block: Does not allow the selected device_class to connect to the device. FullAccess: Allows the selected device_class to connect to the device.</td>
</tr>
<tr>
<td>device_class</td>
<td>AndroidUSB: A portable device running Android OS, like a smartphone or tablet. <strong>Note:</strong> An Android device could connect and be identified as Android, Still Image, or Windows Portable Device (WPD). When blocking Android devices, consider also blocking Still Image and Windows Portable Devices.</td>
</tr>
<tr>
<td>iOS: An Apple portable device running iOS, like an iPhone or iPad.</td>
<td></td>
</tr>
</tbody>
</table>
### Field Name | Description
--- | ---
**Note:** iOS devices will not charge when Device Control is enabled and set to Block, unless the iOS device is powered off. Apple includes their charging capability within functions of the device that are required for our iOS device blocking capability.  
- StillImage: The device class that includes scanners, digital cameras, multi-mode video cameras with frame capture, and frame grabbers.  
- USBCDDVDRW: A USB optical drive.  
- USBDrive: A USB hard drive or USB flash drive.  
- VMWareMount: The VMware USB Passthrough that allows a VMware virtual machine client to access USB devices connected to the host.  
- WPD - A portable device that uses the Microsoft Windows Portable Device (WPD) driver technology, such as a mobile phone, digital camera, and portable media players.  
- exclusion_list: Device Control exclusions that allow full access or block connecting a USB device. The vendor_id is required.  
- comment: Optional information about why the exclusion was added.  
- control_mode:  
  - Block: Blocks the specified vendor_id from connecting to the device.  
  - FullAccess: Allows the specified vendor_id to connect to the device.  
- date_added: The date and time the Device Control exclusion was added to the policy.  
- product_id: Some manufacturers provide a unique identifier for each USB product they make. This information is optional.  
- serial_number: Some manufacturers provide a unique serial number for each USB device they make. This information is optional.  
- vendor_id: The unique identifier for the manufacturer of the USB device.

| file_exclusions | The Policy Safe List identifies file exclusions specific to the policy, and any devices assigned to the policy will allow the excluded files to run.  
| --- | ---
| av_industry: |  
  - false: The file hash has not been identified by the anti-virus industry.  
  - true: The file hash has been identified by the anti-virus industry.  
| category_id: | The Category selected when adding the file to the policy Safe List.  
<p>| 1 = None |<br />
| 2 = Admin Tool |</p>
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 = Internal Application</td>
<td>cloud_score: The Cylance Score displayed in the Console. The score can go up to 100.</td>
</tr>
<tr>
<td>4 = Commercial Software</td>
<td>file_hash: The SHA256 has for the file. This information is required.</td>
</tr>
<tr>
<td>5 = Operating System</td>
<td>file_name: The name of the file. This is &quot;null&quot; if the filename is not available.</td>
</tr>
<tr>
<td>6 = Drivers</td>
<td>file_type: The file scanner type. Should always return 1 for executable.</td>
</tr>
<tr>
<td>7 = Security Software</td>
<td>infinity: The Cylance Cloud score. This is &quot;null&quot; if no score is available.</td>
</tr>
<tr>
<td></td>
<td>md5: The MD5 hash for the file. This information is optional.</td>
</tr>
<tr>
<td></td>
<td>reason: The reason for adding the file to the policy Safe List. The reason must be added when creating the file exclusion.</td>
</tr>
<tr>
<td></td>
<td>research_class_id: The Cylance threat classification. If &quot;infinity&quot; is null, then the research_class_id is not available.</td>
</tr>
<tr>
<td></td>
<td>research_subclass_id: Some threat classification have sub-classes to help administrators determine if a file should be blocked or allowed to run. See &quot;Threat Classifications&quot; on page 327 for more information.</td>
</tr>
<tr>
<td>filetype_actions</td>
<td>The Auto Quarantine of Unsafe (threat_files) and Abnormal (suspicious_files). (Policy &gt; File Actions)</td>
</tr>
<tr>
<td>actions - The setting to enable or disable Auto Quarantine and Auto Upload.</td>
<td>file_type - The only option is &quot;executable&quot;.</td>
</tr>
<tr>
<td>0 - AutoQuarantine OFF and AutoUpload OFF</td>
<td>suspicious_files - Abnormal files</td>
</tr>
<tr>
<td>1 - AutoQuarantine ON and AutoUpload OFF</td>
<td>threat_files - Unsafe files</td>
</tr>
<tr>
<td>2 - AutoQuarantine OFF and AutoUpload ON</td>
<td></td>
</tr>
<tr>
<td>3 - AutoQuarantine ON and AutoUpload ON</td>
<td></td>
</tr>
<tr>
<td>logpolicy</td>
<td>The Agent log file settings.</td>
</tr>
<tr>
<td>log_upload - The setting to enable or disable uploading log files.</td>
<td>null - Disabled</td>
</tr>
<tr>
<td>1 - Enabled</td>
<td></td>
</tr>
</tbody>
</table>

Cylance User API Guide, v2.0 rev24, February 2020 | 84
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxlogsize</td>
<td>The maximum file size (in MB) for a single log file.</td>
</tr>
<tr>
<td>retentiodays</td>
<td>The number of days to save log files. Log files older than the set number of days will be deleted.</td>
</tr>
<tr>
<td>memoryviolation_actions</td>
<td>The Violation Types for Memory Protection.</td>
</tr>
<tr>
<td></td>
<td>All Violation Types must be included in the Request.</td>
</tr>
<tr>
<td></td>
<td>- memory_violations:</td>
</tr>
<tr>
<td></td>
<td>• Isassread (LSASS Read) - Memory belonging to the Windows Local Security Authority process has been accessed in a manner that indicates an attempt to obtain users' passwords.</td>
</tr>
<tr>
<td></td>
<td>• outofprocessallocation (Remote Allocation of Memory) - A process has allocated memory in another process. Most allocations will only occur within the same process. This generally indicates an attempt to inject code or data into another process, which may be a step in reinforcing a malicious presence on a system.</td>
</tr>
<tr>
<td></td>
<td>• outofprocessapc (Remote APC Scheduled) - A process has diverted the execution of another process’s thread. This is generally used by an attacker to activate a malicious presence that has been injected into another process.</td>
</tr>
<tr>
<td></td>
<td>• outofprocessCreatethread (Remote Thread Creation) - A process has created a new thread in another process. A process's threads are usually only created by that same process. This is generally used by an attacker to activate a malicious presence that has been injected into another process.</td>
</tr>
<tr>
<td></td>
<td>• outofprocessmap (Remote Mapping of Memory) - A process has introduced code and/or data into another process. This may indicate an attempt to begin executing code in another process and thereby reinforce a malicious presence.</td>
</tr>
<tr>
<td></td>
<td>• outofprocessoverwritecode (Remote Overwrite Code) - A process has modified executable memory in another process. Under normal conditions, executable memory will not be modified, especially by another process. This usually indicates an attempt to divert execution in another process.</td>
</tr>
<tr>
<td></td>
<td>• outofprocessunmapmemory (Remote Unmap of Memory) - A process has removed a Windows executable from the memory of another process. This may indicate an intent to replace the executable image with a modified copy for the purpose of diverting execution.</td>
</tr>
<tr>
<td></td>
<td>• outofprocesswrite (Remote Write to Memory) - A process has modified memory in another process. This is usually an attempt to store code or data in previously allocated memory (see OutOfProcessAllocation), but it is possible that an attacker is trying to overwrite existing memory in order to divert execution for a malicious purpose.</td>
</tr>
<tr>
<td></td>
<td>• outofprocesswritepe (Remote Write PE to Memory) - A process has modified memory in another process to contain an</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>executable image. Generally, this indicates that an attacker is attempting to execute code without first writing that code to disk.</td>
</tr>
<tr>
<td></td>
<td>overwritecode (Overwrite Code) - The code residing in a process's memory has been modified using a technique that may indicate an attempt to bypass Data Execution Prevention (DEP).</td>
</tr>
<tr>
<td></td>
<td>stackpivot (Stack Pivot) - The stack for a thread has been replaced with a different stack. Generally, the system will only allocate a single stack for a thread. An attacker would use a different stack to control execution in a way that is not blocked by Data Execution Prevention (DEP).</td>
</tr>
<tr>
<td></td>
<td>stackprotect (Stack Protect) - The memory protection of a thread’s stack has been modified to enable execution permission. Stack memory should not be executable, so usually this means that an attacker is preparing to run malicious code stored in stack memory as part of an exploit, an attempt which would otherwise be blocked by Data Execution Prevention (DEP).</td>
</tr>
<tr>
<td></td>
<td>memory_violation_ext:</td>
</tr>
<tr>
<td></td>
<td>dylidinjection (DYLD Injections) - An environment variable has been set that will cause a shared library to be injected into a launched process. Attacks can modify the plist of applications like Safari or replace applications with bash scripts, causing their modules to be loaded automatically when an application starts.</td>
</tr>
<tr>
<td></td>
<td>maliciouspayload (Malicious Payload) - A generic shellcode and payload detection associated with exploitation has been detected.</td>
</tr>
<tr>
<td></td>
<td>trackdataread (RAM Scraping) - A process is trying to read valid magnetic stripe track data from another process. Typically related to point of sale systems (POS).</td>
</tr>
<tr>
<td></td>
<td>zeroallocate (Zero Allocate) - A null page has been allocated. The memory region is typically reserved, but in certain circumstances, it can be allocated. Attacks can use this to setup privilege escalation by taking advantage of some known null de-reference exploit, typically in the kernel.</td>
</tr>
<tr>
<td></td>
<td>memory_exclusion_list: The executable files to exclude from Memory Protection. This must be a relative path to the excluded executable file. Example: \temp</td>
</tr>
<tr>
<td>policy</td>
<td>Various policy settings are contained within this section. All policy settings must be included in the Request.</td>
</tr>
<tr>
<td></td>
<td>auto_blocking - Setting to Auto Quarantine Unsafe threats.</td>
</tr>
<tr>
<td></td>
<td>0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>1 - Enabled</td>
</tr>
<tr>
<td></td>
<td>auto_delete - Setting to automatically delete quarantined files after a set number of days. If this feature is enabled, set &quot;days_until_deleted&quot;</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>auto_uploading - Setting</td>
<td>automatically upload files that Cylance has not seen before. Cylance will perform an analysis on the file and provide details to assist in manual analysis and triage.</td>
</tr>
<tr>
<td>days_until_deleted - Setting</td>
<td>for the number of days to retain a quarantined file. Quarantined files older than the set number of days will be automatically deleted. The minimum number of days is 14, the maximum number of days is 365. The &quot;auto-delete&quot; setting must be enabled.</td>
</tr>
<tr>
<td>device_control - Setting</td>
<td>Setting to enable or disable the Device Control feature.</td>
</tr>
<tr>
<td>full_disc_scan - Setting</td>
<td>Setting to have Cylance analyze all executable files on disk to detect any dormant threats. This is the Background Threat Detection setting.</td>
</tr>
<tr>
<td>kill_running_threats - Setting</td>
<td>Setting to kill processes and children processes regardless of the state when a threat is detected (EXE or DLL).</td>
</tr>
<tr>
<td>low_confidence_threshold - Setting</td>
<td>Setting to adjust the Cylance Score threshold between Unsafe and Abnormal threats. The default is -600, therefore:</td>
</tr>
<tr>
<td>memory_exploit_detection - Setting</td>
<td>Setting to enable or disable the Memory Protection feature. This affects &quot;memory_violation_actions&quot; (&quot;memory_violations&quot; and &quot;memory_violations_ext&quot;).</td>
</tr>
<tr>
<td>oie_auto_uploading - Setting</td>
<td>Currently not in use.</td>
</tr>
<tr>
<td>optics - Setting</td>
<td>Setting to enable or disable CylanceOPTICS.</td>
</tr>
<tr>
<td>optics_application_control_auto_upload - Setting</td>
<td>Setting to allow the automatic upload of files that Cylance has not seen before.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| uploading of Application Control related Focus Data. | ▪ 0 - Disabled  
▪ 1 - Enabled  
▪ optics_malware_auto_upload - Setting to allow the automatic uploading of Threat related Focus Data.  
▪ 0 - Disabled  
▪ 1 - Enabled  
▪ optics_memory_defense_auto_upload - Setting to allow the automatic uploading of Memory Protection related Focus Data.  
▪ 0 - Disabled  
▪ 1 - Enabled  
▪ optics_script_control_auto_upload - Setting to allow the automatic uploading of Script Control related Focus Data.  
▪ 0 - Disabled  
▪ 1 - Enabled  
▪ optics_sensors_advanced_executable_parsing: Setting to enable recording data fields associated with Portable Executable (PE) files, such as File version, Import functions, and Packer types. This is Enhanced Portable Executable Parsing in the policy settings.  
▪ 0 - Disabled  
▪ 1 - Enabled  
▪ optics_sensors_advanced_powershell_visibility: Setting to enable recording commands, arguments, scripts, and content entered directly into the Powershell Console and the Powershell Integrated Scripting Environment (ISE).  
▪ 0 - Disabled  
▪ 1 - Enabled  
▪ optics_sensors_advanced_wmi_visibility: Setting to enable recording additional Windows Management Instrumentation (WMI) attributes and parameters.  
▪ 0 - Disabled  
▪ 1 - Disabled  
▪ optics_sensors_dns_visibility: Setting to enable recording commands and arguments of commands issued directly or indirectly to the Windows Management Instrumentation (WMI) interpreter.  
▪ 0 - Disabled  
▪ 1 - Enabled  
▪ optics_sensors_enhanced_process_hooking_visibility: Setting to enable recording process information from the Win32 API and Kernel Audit messages to detect forms of process hooking and injection.  
▪ 0 - Disabled  
▪ 1 - Enabled  
▪ optics_sensors_private_network_address_visibility: Setting to enable recording network connections within the RFC 1918 and RFC 3419
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address spaces.</td>
<td></td>
</tr>
</tbody>
</table>
|   • 0 - Disabled  
|   • 1 - Enabled |
| optics_sensors_windows_event_log_visibility: | Setting to enable recording Windows Security Events and their associated attributes. |
|   • 0 - Disabled  
|   • 1 - Enabled |
| optics_set_disk_usage_maximum_fixed: | Setting the maximum amount of device storage reserved for use by CylanceOPTICS, in MB. The minimum value is 500 and the maximum value is 1000. |
|   Example: |
|   ```json |
|       {  |
|         "name": "optics_set_disk_usage_maximum_fixed",  |
|         "value": "1000"  |
|       }  |
|   ``` |
| optics_show_notifications: | Setting to enable or disable Desktop Notifications on the endpoint for CylanceOPTICS events. |
|   • 0 - Disabled  
|   • 1 - Enabled |
| pdf_auto_uploading: |
| powershell_auto_uploading: |
| prevent_service_shutdown: | Setting that protects the Cylance service from being shutdown, either manually or by another process. |
|   • 0 - Disabled  
|   • 1 - Enabled |
| python_auto_uploading: |
| sample_copy_path: | Setting to copy all file samples to a network share (CIFS/SMB).  
|   Example: |
|   ```json |
|       {  |
|         "name": "sample_copy_path",  |
|         "value": "\\server_name\shared_folder"  |
|       }  |
|   ``` |
| scan_exception_list: | Setting to exclude specific folders and subfolders from being scanned by full_disc_scan and watch_for_new_files. Set the value to the absolute path for the excluded files.  
|   Example: |
|   ```json |
|       {  |
|         "name": "scan_exception_list",  |
|         "value": [  |
|             "c:\\temp"  |
|         ]  |
|       }  |
|   ``` |
| scan_max_archive_size: | Setting for the maximum archive file size (in
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB) to be scanned. The value can be 0 to 150. If set to 0, then archive files will not be scanned. Example:</td>
<td>{</td>
</tr>
<tr>
<td>{</td>
<td>&quot;name&quot;: &quot;scan_max_archive_size&quot;,</td>
</tr>
<tr>
<td>&quot;value&quot;: &quot;0&quot; }</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>■ script_control: Setting to enable or disable the Script Control feature. Note: Also set the script_control settings (see below in this table).</td>
<td>• 0 - Disabled</td>
</tr>
<tr>
<td>■ show_notifications: Setting to enable or disable Desktop Notifications on the endpoint for CylancePROTECT events.</td>
<td>• 1 - Enabled</td>
</tr>
<tr>
<td>■ threat_report_limit: The number of threats to upload to the Console. Example:</td>
<td>{</td>
</tr>
<tr>
<td>{</td>
<td>&quot;name&quot;: &quot;threat_report_limit&quot;,</td>
</tr>
<tr>
<td>&quot;value&quot;: &quot;500&quot; }</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>■ trust_files_in_scan_exception_list: Setting to Allow Execution of files in the excluded folders. This is related to the scan_exception_list.</td>
<td>0 - Disabled</td>
</tr>
<tr>
<td>■ watch_for_new_files: Setting to analyze new or modified executable files for threats.</td>
<td>1 - Enabled (Allows execution of files in the excluded folders)</td>
</tr>
<tr>
<td>■ optics_set_disk_usage_maximum_fixed - Setting the maximum amount of device storage reserved for use by CylanceOPTICS, in MB. The minimum amount is 500MB and the maximum is 1000MB.</td>
<td></td>
</tr>
<tr>
<td>■ optics_show_notification: Setting to enable or disable CylanceOPTICS Desktop Notifications on the device.</td>
<td>0 - Disabled</td>
</tr>
<tr>
<td>■ pdf_auto_uploading: Currently not in use.</td>
<td>1 - Enabled</td>
</tr>
<tr>
<td>■ powershell_auto_uploading: Currently not in use.</td>
<td></td>
</tr>
<tr>
<td>■ prevent_service_shutdown - Setting that protects the Cylance service from being shutdown, either manually or by another process.</td>
<td></td>
</tr>
<tr>
<td>■ python_auto_uploading: Currently not in use.</td>
<td></td>
</tr>
<tr>
<td>■ sample_copy_path - Setting to copy all file samples to a network share</td>
<td></td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| (CIFS/SMB). Use the fully qualified path. | Example:\server_name\shared_folder.  
- scan_exception_list - Setting to exclude specific folders and subfolders from being scanned by "full_disc_scan" and "watch_for_new_files". Set the value to the absolute path for the excluded files.  
- scan_max_archive_size - Setting for the maximum archive file size (in MB) to be scanned. The value can be 0 to 150. If set to 0, then archive files will not be scanned.  
- script_control - Setting to enable or disable the Script Control feature.  
  - 0 - Disabled  
  - 1 - Enabled  
- show_notifications - Setting to enable or disable Desktop Notifications on the device.  
  - 0 - Disabled  
  - 1 - Enabled  
- threat_report_limit - The number of threats to upload to the Console.  
- trust_files_in_scan_exception_list - Setting to Allow Execution of files in the excluded folders. This is related to the scan_exception_list.  
  - 0 - Disabled  
  - 1 - Enabled (Allows execution of files in the excluded folders)  
- watch_for_new_files - Setting to analyze new or modified executable files for threats. |
| policy_id                  | The unique identifier for the policy.                                                                                                       |
| policy_name                | The name of the policy.                                                                                                                     |
| policy_utc_timestamp       | The date and time the policy was created (in UTC).                                                                                           |
| script_control             | Script Control settings in a policy.                                                                                                        |
|                           | - activescript_settings:  
  - control_mode: Allows or blocks ActiveScript usage.  
    - Allow  
    - Block  
- global_settings:  
  - allowed_folders: Specifies folder exclusions, including subfolders, for Script Control. Specifies a relative path.  
  - control_mode: Allows or blocks ActiveScript and PowerShell usage with Agent 1370 or lower.  
    - Allow  
    - Block  
- macro_settings:  
  - control_mode: Allows or blocks Macro usage. Microsoft Office macros use Visual Basic for Applications (VBA). |
Create Policy

Create a policy.

Service Endpoint:

- /policies/v2

Example: https://protectapi.cylance.com/policies/v2

Method:

- HTTP/1.1 POST

Request Header:

- Content-Type: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the policy:create scope encoded.

Request:

Note: The following example creates a policy with all features disabled (blank policy), except for Execution Control (which is always enabled).

```json
{
    "user_id": "a2c0ac7a-a63d-4583-b646-ae10db9c9769",
    "policy": {
        "policy": [
            {
                "name": "auto_blocking",
                "value": "0"
            }
        ]
    }
}
```
"name": "auto_uploading",
"value": "0"
},
{ "name": "threat_report_limit",
"value": "500"
},
{ "name": "low_confidence_threshold",
"value": "-600"
},
{ "name": "full_disc_scan",
"value": "0"
},
{ "name": "watch_for_new_files",
"value": "0"
},
{ "name": "memory_exploit_detection",
"value": "0"
},
{ "name": "trust_files_in_scan_exception_list",
"value": "0"
},
{ "name": "logpolicy",
"value": "0"
},
{ "name": "script_control",
"value": "0"
},
{ "name": "prevent_service_shutdown",
"value": "0"
},
{ "name": "scan_max_archive_size",
"value": "0"
},
{ "name": "sample_copy_path",
"value": null
},
{ "name": "kill_running_threats",
"value": "0"
}
Cylance User API Guide, v2.0 rev24, February 2020 | 94
"value": "0"
},
  "name": "optics",
  "value": "0"
},
  "name": "auto_delete",
  "value": "0"
},
  "name": "days_until_deleted",
  "value": "14"
},
  "name": "pdf_auto_uploading",
  "value": "0"
},
  "name": "ole_auto_uploading",
  "value": "0"
},
  "name": "docx_auto_uploading",
  "value": "0"
},
  "name": "python_auto_uploading",
  "value": "0"
},
  "name": "autoit_auto_uploading",
  "value": "0"
},
  "name": "powershell_auto_uploading",
  "value": "0"
},
  "name": "custom_thumbprint",
  "value": null
},
  "name": "scan_exception_list",
  "value": []
},
  "name": "optics_show_notifications",
  "value": "0"
}]
"device_control": {
"configurations": [
    {
        "device_class": "AndroidUSB",
        "control_mode": "FullAccess"
    },
    {
        "device_class": "iOS",
        "control_mode": "FullAccess"
    },
    {
        "device_class": "StillImage",
        "control_mode": "FullAccess"
    },
    {
        "device_class": "USBCDDVDRW",
        "control_mode": "FullAccess"
    },
    {
        "device_class": "USBDrive",
        "control_mode": "FullAccess"
    },
    {
        "device_class": "VMWareMount",
        "control_mode": "FullAccess"
    },
    {
        "device_class": "WPD",
        "control_mode": "FullAccess"
    }
],
"exclusion_list": []
],
"memoryviolation_actions": {
    "memory_violations": [
        {
            "violation_type": "lsassread",
            "action": "Alert"
        },
        {
            "violation_type": "outofprocessunmapmemory",
            "action": "Alert"
        },
        {
            "violation_type": "stackpivot",
            "action": "Alert"
        },
        {
            "violation_type": "stackprotect",
            "action": "Alert"
        },
        {
            "violation_type": "outofprocessoverwritecode",
        }
    ]}
"action": "Alert"
},
{
  "violation_type": "outofprocesscreatethread",
  "action": "Alert"
},
{
  "violation_type": "overwritecode",
  "action": "Alert"
},
{
  "violation_type": "outofprocesswritepe",
  "action": "Alert"
},
{
  "violation_type": "outofprocessallocation",
  "action": "Alert"
},
{
  "violation_type": "outofprocessmap",
  "action": "Alert"
},
{
  "violation_type": "outofprocesswrite",
  "action": "Alert"
},
{
  "violation_type": "outofprocessapc",
  "action": "Alert"
}
],
"memory_violations_ext": [
{
  "violation_type": "dyldinjection",
  "action": "Alert"
},
{
  "violation_type": "trackdataread",
  "action": "Alert"
},
{
  "violation_type": "zeroallocate",
  "action": "Alert"
},
{
  "violation_type": "maliciouspayload",
  "action": "Alert"
}
],
"memory_exclusion_list": []
},
"script_control": {

"powershell_settings": {
    "control_mode": "Alert",
    "console_mode": "Alert"
},
"macro_settings": {
    "control_mode": "Alert"
},
"global_settings": {
    "control_mode": "Alert",
    "allowed_folders": []
},
"activescript_settings": {
    "control_mode": "Alert"
},
"filetype_actions": {
    "suspicious_files": [
        {
            "file_type": "executable",
            "actions": "0"
        }
    ],
    "threat_files": [
        {
            "file_type": "executable",
            "actions": "0"
        }
    ]
},
"logpolicy": {
    "retentiondays": "30",
    "log_upload": null,
    "maxlogsize": "100"
},
"file_exclusions": [],
"checksum": "",
"policy_name": "Test Policy"
}

Response:

201 Created

{}
    "policy": [
        {
            "name": "auto_blocking",
            "value": "0"
        },
        {
            "name": "auto_uploading",
            "value": "0"
        }
    ]
"value": "0",
},
{ "name": "optics_set_disk_usage_maximum_fixed",
  "value": "1000"
},
{ "name": "optics_malware_auto_upload",
  "value": "0"
},
{ "name": "optics_memory_defense_auto_upload",
  "value": "0"
},
{ "name": "optics_script_control_auto_upload",
  "value": "0"
},
{ "name": "optics_application_control_auto_upload",
  "value": "0"
},
{ "name": "optics_sensors_dns_visibility",
  "value": "0"
},
{ "value": "0",
  "name": "optics_sensors_private_network_address_visibility"
},
{ "name": "optics_sensors_windows_event_log_visibility",
  "value": "0"
},
{ "name": "optics_sensors_advanced_powershell_visibility",
  "value": "0"
},
{ "name": "optics_sensors_advanced_wmi_visibility",
  "value": "0"
},
{ "name": "optics_sensors_advanced_executable_parsing",
  "value": "0"
},
{ "name": "optics_sensors_enhanced_process_hooking_visibility",
  "value": "0"
},
{ "name": "device_control",
  "value": "0"
}
"value": "0"
},
{
   "name": "optics",
   "value": "0"
},
{
   "name": "auto_delete",
   "value": "0"
},
{
   "name": "days_until_deleted",
   "value": "14"
},
{
   "name": "pdf_auto_uploading",
   "value": "0"
},
{
   "name": "ole_auto_uploading",
   "value": "0"
},
{
   "name": "docx_auto_uploading",
   "value": "0"
},
{
   "name": "python_auto_uploading",
   "value": "0"
},
{
   "name": "autoit_auto_uploading",
   "value": "0"
},
{
   "name": "powershell_auto_uploading",
   "value": "0"
},
{
   "name": "custom_thumbprint",
   "value": null
},
{
   "name": "scan_exception_list",
   "value": []
},
{
   "name": "optics_show_notifications",
   "value": "0"
}
],
"device_control": {
"configurations": [  
  {  
    "device_class": "AndroidUSB",  
    "control_mode": "FullAccess"  
  },  
  {  
    "device_class": "iOS",  
    "control_mode": "FullAccess"  
  },  
  {  
    "device_class": "StillImage",  
    "control_mode": "FullAccess"  
  },  
  {  
    "device_class": "USBCDDVDRW",  
    "control_mode": "FullAccess"  
  },  
  {  
    "device_class": "USBDrive",  
    "control_mode": "FullAccess"  
  },  
  {  
    "device_class": "VMWareMount",  
    "control_mode": "FullAccess"  
  },  
  {  
    "device_class": "WPD",  
    "control_mode": "FullAccess"  
  }  
],  
"exclusion_list": []  
},  
"memoryviolation_actions": {  
  "memory_violations": [ 
    {  
      "violation_type": "lsassread",  
      "action": "Alert"  
    },  
    {  
      "violation_type": "outofprocessunmapmemory",  
      "action": "Alert"  
    },  
    {  
      "violation_type": "stackpivot",  
      "action": "Alert"  
    },  
    {  
      "violation_type": "stackprotect",  
      "action": "Alert"  
    }  
  ]  
}
"action": "Alert"
],
"violation_type": "outofprocesscreatepathread",
"action": "Alert"
},
{ "violation_type": "overwritecode",
"action": "Alert"
},
{ "violation_type": "outofprocesswritepe",
"action": "Alert"
},
{ "violation_type": "outofprocessallocation",
"action": "Alert"
},
{ "violation_type": "outofprocessmap",
"action": "Alert"
},
{ "violation_type": "outofprocesswrite",
"action": "Alert"
},
{ "violation_type": "outofprocessapc",
"action": "Alert"
}
],
"memory_violations_ext": [
{ "violation_type": "dyldinjection",
"action": "Alert"
},
{ "violation_type": "trackdataread",
"action": "Alert"
},
{ "violation_type": "zeroallocate",
"action": "Alert"
},
{ "violation_type": "maliciouspayload",
"action": "Alert"
}
],
"memory_exclusion_list": []
},
"script_control": {
400 Bad Request - Returned for the following reasons:

- The User create request was empty.
- The Tenant ID cannot be retrieved from the JWT Token.
- The User's email address specified is not a proper email address.
- The User application role specified is not one of the accepted values.
- The zones array is empty when the User application role is not Administrator.
- The email provided is already in use.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
500 Internal Server Error - An unforeseeable error has occurred.

**Request JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>checksum</td>
<td>Required when creating a policy. Use an empty value.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>&quot;checksum&quot;: &quot;&quot;</td>
</tr>
<tr>
<td>device_control</td>
<td>Device Control allows or blocks access to USB mass storage devices.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: device_control must be enabled under policy.</td>
</tr>
<tr>
<td>control_mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Block: Blocks the USB device from connecting to the endpoint.</td>
</tr>
<tr>
<td></td>
<td>■ FullAccess: Allows the USB device to connect to the endpoint.</td>
</tr>
<tr>
<td>device_class</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: All device_class entries must be included in the request.</td>
</tr>
<tr>
<td></td>
<td>■ AndroidUSB: A portable device running Android OS, like a smartphone or a tablet.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: An Android device could connect and be identified as Android, Still Image, or Windows Portable Device. If you want to block Android devices, consider blocking Still Image and Windows Portable Devices as well.</td>
</tr>
<tr>
<td></td>
<td>■ iOS: An Apple portable device running iOS, like an iPhone or iPad.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: iOS devices will not charge when Device Control is enabled and set to Block, unless the Apple device is powered off. Apple includes their charging capability within functions of the device that are required for our iOS device blocking capability. Non-Apple devices do not bundle their charging capability in this manner and are not impacted.</td>
</tr>
<tr>
<td></td>
<td>■ StillImage: The device class containing scanners, digital cameras, multi-mode video cameras with frame capture, and frame grabbers.</td>
</tr>
<tr>
<td></td>
<td>■ USBCDDVDWRW: A USB optical drive.</td>
</tr>
<tr>
<td></td>
<td>■ USBDrive: A USB hard drive or USB flash drive.</td>
</tr>
<tr>
<td></td>
<td>■ VMWareMount: VMware USB Passthrough, which allows a VMware virtual machine client to access USB devices connected to the host.</td>
</tr>
<tr>
<td></td>
<td>■ WPD: Windows Portable Device, which uses the Microsoft Windows Portable Device driver technology, such as mobile phones, digital cameras, and portable media players.</td>
</tr>
<tr>
<td>exclusion_list</td>
<td></td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td><strong>Device Control Exclusion List</strong> allows or blocks access to specific USB mass storage devices.</td>
</tr>
<tr>
<td></td>
<td>▪ <strong>comment</strong>: Add details about the exclusion. This information is optional.</td>
</tr>
<tr>
<td></td>
<td>▪ <strong>control_mode</strong>: Allows or blocks the specific USB mass storage device.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Block</strong>: Blocks the USB mass storage device from connecting to the endpoint.</td>
</tr>
<tr>
<td></td>
<td>- <strong>FullAccess</strong>: Allows the USB mass storage device to connect to the endpoint.</td>
</tr>
<tr>
<td></td>
<td>▪ <strong>product_id</strong>: The product identifier for the USB mass storage device. This information is optional.</td>
</tr>
<tr>
<td></td>
<td>▪ <strong>serial_number</strong>: The serial number for the USB mass storage device. This information is optional.</td>
</tr>
<tr>
<td></td>
<td>▪ <strong>vendor_id</strong>: The vendor identifier for the USB mass storage device. This information is required.</td>
</tr>
<tr>
<td><strong>Note</strong>:</td>
<td>One way to find the Vendor ID for a USB mass storage device is to enable Device Control in a policy, assign that policy to an endpoint, then attach the USB mass storage device to the endpoint. You can view External Device logs in the Cylance Console, on the Protection page or the Device Details page (External Devices tab).</td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong>:</td>
</tr>
<tr>
<td></td>
<td>&quot;exclusion_list&quot;: [</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;vendor_id&quot;: &quot;1234&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;comment&quot;: &quot;Test device control exclusion&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;serial_number&quot;: 987654321,</td>
</tr>
<tr>
<td></td>
<td>&quot;product_id&quot;: &quot;5678&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;control_mode&quot;: &quot;FullAccess&quot;</td>
</tr>
<tr>
<td></td>
<td>]</td>
</tr>
<tr>
<td><strong>file_exclusions</strong></td>
<td>Adds file exclusions to the Policy Safe List, under File Actions. Policy Safe List are file exclusions specific to the policy, and any endpoints assigned to the policy will allow the excluded files to run.</td>
</tr>
<tr>
<td></td>
<td>▪ <strong>category_id</strong>: A list of categories to identify the type of file. This information is optional.</td>
</tr>
<tr>
<td></td>
<td>- 1 - None</td>
</tr>
<tr>
<td></td>
<td>- 2 - AdminTool</td>
</tr>
<tr>
<td></td>
<td>- 3 - InternalApplication</td>
</tr>
<tr>
<td></td>
<td>- 4 - CommercialSoftware</td>
</tr>
<tr>
<td></td>
<td>- 5 - OperatingSystem</td>
</tr>
<tr>
<td></td>
<td>- 6 - Drivers</td>
</tr>
<tr>
<td></td>
<td>- 7 - SecuritySoftware</td>
</tr>
<tr>
<td></td>
<td>▪ <strong>file_hash</strong>: The SHA256 hash for the file. This information is required.</td>
</tr>
<tr>
<td></td>
<td>▪ <strong>file_name</strong>: The name of the file being excluded. This information is optional.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>md5:</td>
<td>The MD5 hash for the file. This information is optional.</td>
</tr>
<tr>
<td>reason:</td>
<td>The reason the file was excluded. This information is required.</td>
</tr>
<tr>
<td>Example:</td>
<td>&quot;file_exclusions&quot;: [</td>
</tr>
<tr>
<td></td>
<td>&quot;reason&quot;: &quot;Test Exclusion&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;category_id&quot;: &quot;2&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;md5&quot;: &quot;d41d8cd98f00b204e9800998ecf8427e&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;file_hash&quot;: &quot;bf17366ee3bb8068a9ad70f9c9e68496e7e311a055bf4ffeef53cc5ccce52&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;file_name&quot;: &quot;filename&quot;</td>
</tr>
<tr>
<td>]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>filetype_actions</th>
<th>The Auto-Quarantine of Unsafe (threat_files) and Abnormal (suspicious_files).</th>
</tr>
</thead>
<tbody>
<tr>
<td>actions:</td>
<td>Set Auto-Quarantine and Auto-Upload to Enable or Disable</td>
</tr>
<tr>
<td>0</td>
<td>Auto-Quarantine OFF, Auto-Upload OFF</td>
</tr>
<tr>
<td>1</td>
<td>Auto-Quarantine ON, Auto-Upload OFF</td>
</tr>
<tr>
<td>2</td>
<td>Auto-Quarantine OFF, Auto-Upload OFF</td>
</tr>
<tr>
<td>3</td>
<td>Auto-Quarantine ON, Auto-Upload ON</td>
</tr>
<tr>
<td>file_type:</td>
<td>The only option is &quot;executable&quot;</td>
</tr>
<tr>
<td>suspicious_files:</td>
<td>Abnormal files</td>
</tr>
<tr>
<td>threat_files:</td>
<td>Unsafe files</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>logpolicy</th>
<th>The Agent log file settings.</th>
</tr>
</thead>
<tbody>
<tr>
<td>log_upload:</td>
<td>The setting to enable or disable uploading Agent log files.</td>
</tr>
<tr>
<td>null</td>
<td>Disabled</td>
</tr>
<tr>
<td>1</td>
<td>Enabled</td>
</tr>
<tr>
<td>maxlogsize:</td>
<td>The maximum file size (in MB) for a single Agent log file.</td>
</tr>
<tr>
<td>retentiondays:</td>
<td>The number of days to save Agent log files. Log files older than the set number of days will be deleted.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>memoryviolation_actions</th>
<th>The Violation Types for Memory Protection.</th>
</tr>
</thead>
<tbody>
<tr>
<td>All memoryViolations and memoryViolations_ext entries must be included in the Request.</td>
<td></td>
</tr>
<tr>
<td>memoryViolations</td>
<td></td>
</tr>
<tr>
<td>Isassread (LSASS Read): Memory belonging to the Windows Local Security Authority process has been accessed in a manner that indicates an attempt to obtain Users’ passwords.</td>
<td></td>
</tr>
<tr>
<td>outofprocessallocation (Remote Allocation of Memory): A process has allocated memory in another process. Most allocations will only occur</td>
<td></td>
</tr>
</tbody>
</table>
within the same process. This generally indicates an attempt to inject code or data into another process, which may be a first step in reinforcing a malicious presence on a system.

- outofprocessapc (Remote APC Scheduled): A process has diverted the execution of another process's thread. This is generally used by an attacker to activate a malicious presence that has been injected into another process.

- outofprocesscreatethread (Remote Thread Creation): A process has created a new thread in another process. A process's threads are usually only created by that same process. This is generally used by an attacker to activate a malicious presence that has been injected into another process.

- outofprocessmap (Remote Mapping of Memory): A process has introduced code and/or data into another process. This may indicate an attempt to begin executing code in another process and thereby reinforce a malicious presence.

- outofprocessoverwritecode (Remote Overwrite Code): A process has modified executable memory in another process. Under normal conditions, executable memory will not be modified, especially by another process. This usually indicates an attempt to divert execution in another process.

- outofprocessunmapmemory (Remote Unmap of Memory): A process has removed a Windows executable from the memory of another process. This may indicate an intent to replace the executable image with a modified copy for the purpose of diverting execution.

- outofprocesswrite (Remote Write to Memory): A process has modified memory in another process. This is usually an attempt to store code or data in previously allocated memory (see OutOfProcessAllocation), but it is possible that an attacker is trying to overwrite existing memory in order to divert execution for a malicious purpose.

- outofprocesswritepe (Remote Write PE to Memory): A process has modified memory in another process to contain an executable image. Generally, this indicates that an attacker is attempting to execute code without first writing that code to disk.

- overwritecode (Overwrite Code): The code residing in a process's memory has been modified using a technique that may indicate an attempt to bypass Data Execution Prevention (DEP).

- stackpivot (Stack Pivot): The stack for a thread has been replaced with a different stack. Generally, the system will only allocate a single stack for a thread. An attacker would use a different stack to control execution in a way that is not blocked by Data Execution Prevention (DEP).

- stackprotect (Stack Protect): The memory protection of a thread's stack has been modified to enable execution permission. Stack memory should not be executable, so usually this means that an attacker is preparing to run malicious code stored in stack memory as part of an exploit, an attempt which would otherwise be blocked by Data Execution Prevention (DEP).

```
memory_violations_ext
```

- dyldinjection (DYLD Injection): An environment variable has been set that
### Field Name | Description
--- | ---
 | will cause a shared library to be injected into a launched process. Attacks can modify the plist of applications like Safari or replace applications with bash scripts, causing their modules to be loaded automatically when an application starts.  
- maliciouspayload (Malicious Payload): A generic shellcode and payload detection associated with exploitation has been detected.  
- trackdataread (RAM Scraping): A process is trying to read valid magnetic stripe track data from another process. Typically related to point of sale systems (POS).  
- zeroallocate (Zero Allocate): A null page has been allocated. The memory region is typically reserved, but in certain circumstances, it can be allocated. Attacks can use this to setup privilege escalation by taking advantage of some known null de-reference exploit, typically in the kernel.

**memory_exclusion_list**

The executable files to exclude from Memory Protection. This must be a relative path to the excluded executable file.

Example:

```
"memory_exclusion_list": [  
"\\temp"
]
```

| policy | Various policy settings are contained within this section. Some policy settings are enabled under policy and configured in a different section, like device_control and logpolicy.  
- auto_blocking: Enables or disables the Auto Quarantine setting for Unsafe and Abnormal files.  
  **Note:** filetype_actions must also be set for Unsafe (threat_files) and Abnormal (suspicious_files) files.  
  - 0 - Disabled  
  - 1 - Enabled  
- auto_delete: Setting to automatically delete quarantined files after a set number of days.  
  **Note:** If this feature is enabled, set days_until_deleted for the number of days to retain a quarantined file.  
  - 0 - Disabled  
  - 1 - Enabled  
- auto_uploading: Setting to automatically upload files that Cylance has not seen before. Cylance will perform an analysis on the file and provide details to assist in manual analysis and triage.  
  **Note:** filetype_actions must also be set for Unsafe (threat_files) and Abnormal (suspicious_files) files for Auto-Upload.  
  - 0 - Disabled  

Cylance User API Guide, v2.0 rev24, February 2020 | 109
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>autoit_auto_uploading</td>
<td></td>
</tr>
<tr>
<td>custom_thumbprint</td>
<td></td>
</tr>
</tbody>
</table>
| days_until_deleted | Setting for the number of days to retain a quarantined file. Quarantined files older than the set number of days will be automatically deleted. The minimum number of days is 14, the maximum number of days is 365. **Note:** To use the *days_until_deleted* feature, the *auto_delete* setting must be enabled.  
Example:  
```json  
{  
  "name": "days_until_deleted",  
  "value": "14"  
}  
``` |
| device_control | Setting to enable or disable the Device Control feature. Use *device_control* settings to Block or give FullAccess to specific types of USB mass storage devices. See *device_control* towards the beginning of this table.  
- 0 - Disabled  
- 1 - Enabled |
| docx_auto_uploading | |
| full_disc_scan | Setting to have Cylance analyze all executable files on disk to detect any dormant threats. This is the Background Threat Detection (BTD) setting.  
- 0 - Disabled  
- 1 - Run Recurring (performs a scan every nine days)  
- 2 - Run Once (runs a full disk scan upon installation only) |
| kill_running_threats | Setting to kill processes and child processes regardless of the state when a threat is detected (EXE or DLL).  
- 0 - Disabled  
- 1 - Enabled |
| logpolicy | This setting is not used. |
| low_confidence_threshold | Setting to adjust the Cylance Score threshold between Unsafe and Abnormal files. The default is -600, therefore:  
- A Cylance Score of -600 to -1000 is Unsafe.  
- A Cylance Score of 0 to -599 is Abnormal.  
- A Cylance Score greater than 0 is Safe.  
Example:  
```json  
{  
  "name": "low_confidence_threshold",  
  "value": "-600"  
}  
``` |
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>memory_exploit_detection</td>
<td>Setting to enable or disable the Memory Protection feature.</td>
</tr>
<tr>
<td>Note:</td>
<td>Also set the memoryviolation_actions (memory_violations, memory_violations_ext, and memory_exclusion_list).</td>
</tr>
<tr>
<td></td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>ole_auto_uploading</td>
<td></td>
</tr>
<tr>
<td>opts_c:</td>
<td>Setting to enable or disable CylanceOPTICS.</td>
</tr>
<tr>
<td>Note:</td>
<td>Also set the other CylanceOPTICS (opts_) settings.</td>
</tr>
<tr>
<td></td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>opts_application_control_auto_upload</td>
<td>Setting to allow the automatic uploading of Application Control related Focus Data.</td>
</tr>
<tr>
<td></td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>opts_malware_auto_upload</td>
<td>Setting to allow the automatic uploading of Threat related Focus Data.</td>
</tr>
<tr>
<td></td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>opts_memory_defense_auto_upload</td>
<td>Setting to allow the automatic uploading of Memory Protection related Focus Data.</td>
</tr>
<tr>
<td></td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>opts_script_control_auto_upload</td>
<td>Setting to allow the automatic uploading of Script Control related Focus Data.</td>
</tr>
<tr>
<td></td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>opts_sensors_advanced_executable_parsing</td>
<td>Setting to enable recording data fields associated with Portable Executable (PE) files, such as File version, Import functions, and Packer types. This is Enhanced Portable Executable Parsing in the policy settings.</td>
</tr>
<tr>
<td></td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>opts_sensors_advanced_powershell_visibility</td>
<td>Setting to enable recording commands, arguments, scripts, and content entered directly into the Powershell Console and the Powershell Integrated Scripting Environment (ISE).</td>
</tr>
<tr>
<td></td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>opts_sensors_advanced_wmi_visibility</td>
<td>Setting to enable recording additional Windows Management Instrumentation (WMI) attributes and parameters.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>options_sensors_dns_visibility</td>
<td>Setting to enable recording commands and arguments of commands issued directly or indirectly to the Windows Management Instrumentation (WMI) interpreter.</td>
</tr>
<tr>
<td>options_sensors_enhanced_process_hooking_visibility</td>
<td>Setting to enable recording process information from the Win32 API and Kernel Audit messages to detect forms of process hooking and injection.</td>
</tr>
<tr>
<td>options_sensors_private_network_address_visibility</td>
<td>Setting to enable recording network connections within the RFC 1918 and RFC 3419 address spaces.</td>
</tr>
<tr>
<td>options_sensors_windows_event_log_visibility</td>
<td>Setting to enable recording Windows Security Events and their associated attributes.</td>
</tr>
<tr>
<td>options_set_disk_usage_maximum_fixed</td>
<td>Setting the maximum amount of device storage reserved for use by CylanceOPTICS, in MB. The minimum value is 500 and the maximum value is 1000.</td>
</tr>
<tr>
<td>Example:</td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;name&quot;: &quot;options_set_disk_usage_maximum_fixed&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;value&quot;: &quot;1000&quot;</td>
</tr>
<tr>
<td>options_show_notifications</td>
<td>Setting to enable or disable Desktop Notifications on the endpoint for CylanceOPTICS events.</td>
</tr>
<tr>
<td>pdf_auto_uploading</td>
<td></td>
</tr>
<tr>
<td>powershell_auto_uploading</td>
<td></td>
</tr>
<tr>
<td>prevent_service_shutdown</td>
<td>Setting that protects the Cylance service from being shutdown, either manually or by another process.</td>
</tr>
<tr>
<td>python_auto_uploading</td>
<td></td>
</tr>
<tr>
<td>sample_copy_path</td>
<td>Setting to copy all file samples to a network share (CIFS/SMB).</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>{ &quot;name&quot;: &quot;sample_copy_path&quot;, &quot;value&quot;: &quot;\server_name\shared_folder&quot; }</td>
<td></td>
</tr>
<tr>
<td>- scan_exception_list: Setting to exclude specific folders and subfolders from being scanned by full_disc_scan and watch_for_new_files. Set the value to the absolute path for the excluded files. Example: { &quot;name&quot;: &quot;scan_exception_list&quot;, &quot;value&quot;: [ &quot;c:temp&quot; ] }</td>
<td></td>
</tr>
<tr>
<td>- scan_max_archive_size: Setting for the maximum archive file size (in MB) to be scanned. The value can be 0 to 150. If set to 0, then archive files will not be scanned. Example: { &quot;name&quot;: &quot;scan_max_archive_size&quot;, &quot;value&quot;: &quot;0&quot; }</td>
<td></td>
</tr>
<tr>
<td>- script_control: Setting to enable or disable the Script Control feature. <strong>Note:</strong> Also set the script_control settings (see below in this table).</td>
<td></td>
</tr>
</tbody>
</table>
| | - 0 - Disabled  
| | - 1 - Enabled  |
| - show_notifications: Setting to enable or disable Desktop Notifications on the endpoint for CylancePROTECT events. |  |
| | - 0 - Disabled  
| | - 1 - Enabled  |
| - threat_report_limit: The number of threats to upload to the Console. Example: { "name": "threat_report_limit", "value": "500" } | |
| - trust_files_in_scan_exception_list: Setting to Allow Execution of files in the excluded folders. This is related to the scan_exception_list. 0 - Disabled  
| | - 1 - Enabled (Allows execution of files in the excluded folders)  |
| - watch_for_new_files: Setting to analyze new or modified executable files for threats. |  |
| | - 0 - Disabled  
<p>| | - 1 - Enabled (Allows analysis of new or modified executable files)  |</p>
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>policy_name</td>
<td>The name of the policy. The name must be unique to your tenant.</td>
</tr>
</tbody>
</table>
| script_control | The policy settings for Script Control.  
  **Note:** *script_control* must be enabled (set to "1") under *policy*. 
  **activescript_settings**  
  - control_mode: Settings for Active Script.  
    - Alert: An alert is sent when an Active Script event occurs. The Active Script is allowed to run.  
    - Block: The Active Script is blocked and an alert is sent. 
  **global_settings**  
  - allowed_folders: The relative path to scripts that are allowed to run when Script Control is enabled. Script Control Folder Exclusions apply to all Agent versions (Agent 1310 or higher).  
    Example:  
    "allowed_folders": [  
      "\\temp_scriptcontrol"  
    ]  
  - control_mode: Setting to enable or disable Script Control for Agent version 1370 or lower. This works for Active Scripts and PowerShell. This does not work for Macros. To use Script Control with Macros, use Agent version 1380 or higher.  
    - Allow: An alert is sent when an Active Script or PowerShell event occurs. The script is allowed to run.  
    - Block: The Active Script or PowerShell is blocked and an alert is sent. 
  **macro_settings**  
  - control_mode: Settings for Microsoft Office Macros.  
    - Alert: An alert is sent when an Office Macro event occurs. The Macro is allowed to run.  
    - Block: The Office Macro is blocked and an alert is sent. 
  **powershell_settings**  
  - console_mode: The PowerShell Console is blocked to prevent PowerShell command usage, including one-liners. To use this feature, the PowerShell *control_mode* must be set to Block.  
  - control_mode:  
    - Alert: An alert is sent when a PowerShell script event occurs. The PowerShell script is allowed to run.  
    - Block: The PowerShell script is blocked and an alert is sent. 

**About disabling Script Control**
For Agent versions 1430 and higher, you can disable Script Control for Active Script, PowerShell, or Macros. Disabling Script Control allows the selected script type to run and does not send an alert to the Console.

To disable Script Control for a specific script type, do not include the script type in the Create Policy API request.

Example: Script Control for Macros is disabled.

```json
"script_control": {
  "powershell_settings": {
    "control_mode": "Block",
    "console_mode": "Block"
  },
  "global_settings": {
    "control_mode": "Alert",
    "allowed_folders": [
      "\temp_scriptcontrol"
    ]
  },
  "activescript_settings": {
    "control_mode": "Alert"
  }
}
```

user_id

The unique ID for the user creating the policy. Only administrators can create policies.

**Note:** To get the user_id, use "Get Users on page 31."

### Response JSON Schema Descriptions

**Note:** This table only covers descriptions not covered in the Request JSON Schema Descriptions table (see previous table).

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>policy_id</td>
<td>The unique identifier for the policy.</td>
</tr>
<tr>
<td>policy_utctimestamp</td>
<td>The date and time (in UTC) when the policy was created.</td>
</tr>
</tbody>
</table>

## Update Policy

Update an existing policy.

**Note:** The request contents for Update Policy are similar to Create Policy, except you must include the `policy_id` in the Update Policy request.

**Service Endpoint:**

- `/policies/v2`
Example: https://protectapi.cylance.com/policies/v2

Method:
- HTTP/1.1 PUT

Request Headers:
- Content-Type: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the policy:update scope encoded

Request:

```json
{
    "user_id": "a2c0ac7a-a63d-4583-b646-ae10db1c9769",
    "policy": {
        "policy": [
            {
                "name": "auto_blocking",
                "value": "0"
            },
            {
                "name": "auto_uploading",
                "value": "0"
            },
            {
                "name": "threat_report_limit",
                "value": "500"
            },
            {
                "name": "low_confidence_threshold",
                "value": "-600"
            },
            {
                "name": "full_disc_scan",
                "value": "0"
            },
            {
                "name": "watch_for_new_files",
                "value": "0"
            },
            {
                "name": "memory_exploit_detection",
                "value": "0"
            },
            {
                "name": "trust_files_in_scan_exception_list",
                "value": "0"
            },
            {
                "name": "logpolicy",
                "value": "0"
            }
        ]
    }
}
```
"value": "0"
},

"name": "script_control",
"value": "0"
},

"name": "prevent_service_shutdown",
"value": "0"
},

"name": "scan_max_archive_size",
"value": "0"
},

"name": "sample_copy_path",
"value": null
},

"name": "kill_running_threats",
"value": "0"
},

"name": "show_notifications",
"value": "0"
},

"name": "optics_set_disk_usage_maximum_fixed",
"value": "1000"
},

"name": "optics_malware_auto_upload",
"value": "0"
},

"name": "optics_memory_defense_auto_upload",
"value": "0"
},

"name": "optics_script_control_auto_upload",
"value": "0"
},

"name": "optics_application_control_auto_upload",
"value": "0"
},

"name": "optics_sensors_dns_visibility",
"value": "0"
},

"value": "0",}
"name": "optics_sensors_private_network_address_visibility",
},
{ "name": "optics_sensors_windows_event_log_visibility",
  "value": "0"
},
{ "name": "optics_sensors_advanced_powershell_visibility",
  "value": "0"
},
{ "name": "optics_sensors_advanced_wmi_visibility",
  "value": "0"
},
{ "name": "optics_sensors_advanced_executable_parsing",
  "value": "0"
},
{ "name": "optics_sensors_enhanced_process_hooking_visibility",
  "value": "0"
},
{ "name": "device_control",
  "value": "0"
},
{ "name": "optics",
  "value": "0"
},
{ "name": "auto_delete",
  "value": "0"
},
{ "name": "days_until_deleted",
  "value": "14"
},
{ "name": "pdf_auto_uploading",
  "value": "0"
},
{ "name": "ole_auto_uploading",
  "value": "0"
},
{ "name": "docx_auto_uploading",
  "value": "0"
},
{ "name": "python_auto_uploading",}
"value": "0"
],
  "device_control": {
    "configurations": [
      {
        "device_class": "AndroidUSB",
        "control_mode": "FullAccess"
      },
      {
        "device_class": "iOS",
        "control_mode": "FullAccess"
      },
      {
        "device_class": "StillImage",
        "control_mode": "FullAccess"
      },
      {
        "device_class": "USBCDDVDRW",
        "control_mode": "FullAccess"
      },
      {
        "device_class": "USBDrive",
        "control_mode": "FullAccess"
      },
      {
        "device_class": "VMWareMount",
        "control_mode": "FullAccess"
      },
      {
        "device_class": "WPD",
        "control_mode": "FullAccess"

",
"exclusion_list": []
},
"memoryviolation_actions": {
"memory_violations": [
{
"violation_type": "lsassread",
"action": "Alert"
},
{
"violation_type": "outofprocessunmapmemory",
"action": "Alert"
},
{
"violation_type": "stackpivot",
"action": "Alert"
},
{
"violation_type": "stackprotect",
"action": "Alert"
},
{
"violation_type": "outofprocessoverwritecode",
"action": "Alert"
},
{
"violation_type": "outofprocesscreatethread",
"action": "Alert"
},
{
"violation_type": "outofprocessoverwrite",
"action": "Alert"
},
{
"violation_type": "outofprocessallocation",
"action": "Alert"
},
{
"violation_type": "outofprocessmap",
"action": "Alert"
},
{
"violation_type": "outofprocesswrite",
"action": "Alert"
},
{
"violation_type": "outofprocessapc",
"action": "Alert"
}]
}
"memory_violations_ext": [
{
"violation_type": "dyldinjection",
"action": "Alert"
},
{
"violation_type": "trackdataread",
"action": "Alert"
},
{
"violation_type": "zeroallocate",
"action": "Alert"
},
{
"violation_type": "maliciouspayload",
"action": "Alert"
}
],
"memory_exclusion_list": []
},
"script_control": {
"powershell_settings": {
"control_mode": "Alert",
"console_mode": "Alert"
},
"macro_settings": {
"control_mode": "Alert"
},
"global_settings": {
"control_mode": "Alert",
"allowed_folders": []
},
"activescript_settings": {
"control_mode": "Alert"
},
"filetype_actions": {
"suspicious_files": [
{
"file_type": "executable",
"actions": "0"
}
],
"threat_files": [
{
"file_type": "executable",
"actions": "0"
}
]


},
  "logpolicy": {
    "retentiondays": "30",
    "log_upload": null,
    "maxlogsize": "100"
  },
  "file_exclusions": [],
  "checksum": "",
  "policy_name": "Test Policy",
  "policy_id": "d5c6d6a3-0599-4fb5-96bc-0fdc7eacb6ea"
}
}

Response:

204 No Content - Policy updated

400 Bad Request - Returned for the following reasons:

  - The User ID is invalid.
  - The Policy ID is invalid.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

500 Internal Server Error - An unforeseeable error has occurred.

Request JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>checksum</td>
<td>Required when creating a policy. Use an empty value.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>&quot;checksum&quot;: &quot;&quot;</td>
</tr>
<tr>
<td>device_control</td>
<td>Device Control allows or blocks access to USB mass storage devices.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: device_control must be enabled under policy.</td>
</tr>
<tr>
<td>control_mode:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Block: Blocks the USB device from connecting to the endpoint.</td>
</tr>
<tr>
<td></td>
<td>■ FullAccess: Allows the USB device to connect to the endpoint.</td>
</tr>
<tr>
<td>device_class:</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: All device_class entries must be included in the request.</td>
</tr>
<tr>
<td></td>
<td>■ AndroidUSB: A portable device running Android OS, like a smartphone or a</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| iOS | An Apple portable device running iOS, like an iPhone or iPad.  
**Note**: iOS devices will not charge when Device Control is enabled and set to Block, unless the Apple device is powered off. Apple includes their charging capability within functions of the device that are required for our iOS device blocking capability. Non-Apple devices do not bundle their charging capability in this manner and are not impacted. |
| StillImage | The device class containing scanners, digital cameras, multi-mode video cameras with frame capture, and frame grabbers. |
| USBCDDVDRW | A USB optical drive. |
| USBDrive | A USB hard drive or USB flash drive. |
| VMWareMount | VMware USB Passthrough, which allows a VMware virtual machine client to access USB devices connected to the host. |
| WPD | Windows Portable Device, which uses the Microsoft Windows Portable Device driver technology, such as mobile phones, digital cameras, and portable media players. |
| **exclusion_list**: | Device Control Exclusion List allows or blocks access to specific USB mass storage devices. |
| comment | Add details about the exclusion. This information is optional. |
| control_mode | Allows or blocks the specific USB mass storage device. |
| Block | Blocks the USB mass storage device from connecting to the endpoint. |
| FullAccess | Allows the USB mass storage device to connect to the endpoint. |
| product_id | The product identifier for the USB mass storage device. This information is optional. |
| serial_number | The serial number for the USB mass storage device. This information is optional. |
| vendor_id | The vendor identifier for the USB mass storage device. This information is required. |

**Note**: One way to find the Vendor ID for a USB mass storage device is to enable Device Control in a policy, assign that policy to an endpoint, then attach the USB mass storage device to the endpoint. You can view External Device logs in the Cylance Console, on the Protection page or the Device Details page (External Devices tab).  

Example:  
"exclusion_list": [  
  {  
    "vendor_id": "1234",  
    "comment": "Test device control exclusion",  
    "serial_number": "987654321",  
    "product_id": "5678",  
    "control_mode": "FullAccess"  
  }  
]
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file_exclusions</td>
<td>Adds file exclusions to the Policy Safe List, under File Actions. Policy Safe List are file exclusions specific to the policy, and any endpoints assigned to the policy will allow the excluded files to run.</td>
</tr>
<tr>
<td>category_id</td>
<td>A list of categories to identify the type of file. This information is optional.</td>
</tr>
<tr>
<td>- 1 - None</td>
<td></td>
</tr>
<tr>
<td>- 2 - AdminTool</td>
<td></td>
</tr>
<tr>
<td>- 3 - InternalApplication</td>
<td></td>
</tr>
<tr>
<td>- 4 - CommercialSoftware</td>
<td></td>
</tr>
<tr>
<td>- 5 - OperatingSystem</td>
<td></td>
</tr>
<tr>
<td>- 6 - Drivers</td>
<td></td>
</tr>
<tr>
<td>- 7 - SecuritySoftware</td>
<td></td>
</tr>
<tr>
<td>file_hash</td>
<td>The SHA256 hash for the file. This information is required.</td>
</tr>
<tr>
<td>file_name</td>
<td>The name of the file being excluded. This information is optional.</td>
</tr>
<tr>
<td>md5</td>
<td>The MD5 hash for the file. This information is optional.</td>
</tr>
<tr>
<td>reason</td>
<td>The reason the file was excluded. This information is required.</td>
</tr>
<tr>
<td>logpolicy</td>
<td>The Agent log file settings.</td>
</tr>
</tbody>
</table>

Example:

```json
"file_exclusions": [
  {
    "reason": "Test Exclusion",
    "category_id": "2",
    "md5": "d41d8cd98f00b204e9800998ecf8427e",
    "file_hash": "bf17366ee3bb8068a9ad70fc9e68496e7e311a055bf4fffeff53cc5d29ccce52",
    "file_name": "filename"
  }
]
```

filetype_actions | The Auto-Quarantine of Unsafe (threat_files) and Abnormal (suspicious_files).                          |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>actions</td>
<td>Set Auto-Quarantine and Auto-Upload to Enable or Disable</td>
</tr>
<tr>
<td>- 0 - Auto-Quarantine OFF, Auto-Upload OFF</td>
<td></td>
</tr>
<tr>
<td>- 1 - Auto-Quarantine ON, Auto-Upload OFF</td>
<td></td>
</tr>
<tr>
<td>- 2 - Auto-Quarantine OFF, Auto-Upload OFF</td>
<td>Use for suspicious_files when threat_files is set to 3 and Auto-Quarantine for suspicious_files is disabled.</td>
</tr>
<tr>
<td>- 3 - Auto-Quarantine ON, Auto-Upload ON</td>
<td></td>
</tr>
<tr>
<td>file_type</td>
<td>The only option is &quot;executable&quot;</td>
</tr>
<tr>
<td>suspicious_files</td>
<td>Abnormal files</td>
</tr>
<tr>
<td>threat_files</td>
<td>Unsafe files</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>log_upload</td>
<td>The setting to enable or disable uploading Agent log files.</td>
</tr>
<tr>
<td></td>
<td>- null - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>maxlogsize</td>
<td>The maximum file size (in MB) for a single Agent log file.</td>
</tr>
<tr>
<td>retentiondays</td>
<td>The number of days to save Agent log files. Log files older than the set number of days will be deleted.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>memoryviolation_actions</th>
<th>The Violation Types for Memory Protection.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All memory_violations and memory_violations_ext entries must be included in the Request.</td>
</tr>
<tr>
<td>memory_violations</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Isassread (LSASS Read): Memory belonging to the Windows Local Security Authority process has been accessed in a manner that indicates an attempt to obtain Users’ passwords.</td>
</tr>
<tr>
<td></td>
<td>outofprocessallocation (Remote Allocation of Memory): A process has allocated memory in another process. Most allocations will only occur within the same process. This generally indicates an attempt to inject code or data into another process, which may be a first step in reinforcing a malicious presence on a system.</td>
</tr>
<tr>
<td></td>
<td>outofprocessapc (Remote APC Scheduled): A process has diverted the execution of another process's thread. This is generally used by an attacker to activate a malicious presence that has been injected into another process.</td>
</tr>
<tr>
<td></td>
<td>outofprocesscreatethread (Remote Thread Creation): A process has created a new thread in another process. A process’s threads are usually only created by that same process. This is generally used by an attacker to activate a malicious presence that has been injected into another process.</td>
</tr>
<tr>
<td></td>
<td>outofprocessmap (Remote Mapping of Memory): A process has introduced code and/or data into another process. This may indicate an attempt to begin executing code in another process and thereby reinforce a malicious presence.</td>
</tr>
<tr>
<td></td>
<td>outofprocessoverwritecode (Remote Overwrite Code): A process has modified executable memory in another process. Under normal conditions, executable memory will not be modified, especially by another process. This usually indicates an attempt to divert execution in another process.</td>
</tr>
<tr>
<td></td>
<td>outofprocessunmapmemory (Remote Unmap of Memory): A process has removed a Windows executable from the memory of another process. This may indicate an intent to replace the executable image with a modified copy for the purpose of diverting execution.</td>
</tr>
<tr>
<td></td>
<td>outofprocesswrite (Remote Write to Memory): A process has modified memory in another process. This is usually an attempt to store code or data in previously allocated memory (see OutOfProcessAllocation), but it is possible that an attacker is trying to overwrite existing memory in order to divert execution for a malicious purpose.</td>
</tr>
</tbody>
</table>
|                         | outofprocesswritepe (Remote Write PE to Memory): A process has modified memory in another process to contain an executable image. Generally, this
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>indicates that an attacker is attempting to execute code without first writing that code to disk.</td>
</tr>
<tr>
<td></td>
<td>■ overwritecode (Overwrite Code): The code residing in a process’s memory has been modified using a technique that may indicate an attempt to bypass Data Execution Prevention (DEP).</td>
</tr>
<tr>
<td></td>
<td>■ stackpivot (Stack Pivot): The stack for a thread has been replaced with a different stack. Generally, the system will only allocate a single stack for a thread. An attacker would use a different stack to control execution in a way that is not blocked by Data Execution Prevention (DEP).</td>
</tr>
<tr>
<td></td>
<td>■ stackprotect (Stack Protect): The memory protection of a thread’s stack has been modified to enable execution permission. Stack memory should not be executable, so usually this means that an attacker is preparing to run malicious code stored in stack memory as part of an exploit, an attempt which would otherwise be blocked by Data Execution Prevention (DEP).</td>
</tr>
<tr>
<td>memory_violations_ext</td>
<td>■ dyldinjection (DYLD Injection): An environment variable has been set that will cause a shared library to be injected into a launched process. Attacks can modify the plist of applications like Safari or replace applications with bash scripts, causing their modules to be loaded automatically when an application starts.</td>
</tr>
<tr>
<td></td>
<td>■ maliciouspayload (Malicious Payload): A generic shellcode and payload detection associated with exploitation has been detected.</td>
</tr>
<tr>
<td></td>
<td>■ trackdataread (RAM Scraping): A process is trying to read valid magnetic stripe track data from another process. Typically related to point of sale systems (POS).</td>
</tr>
<tr>
<td></td>
<td>■ zeroallocate (Zero Allocate): A null page has been allocated. The memory region is typically reserved, but in certain circumstances, it can be allocated. Attacks can use this to setup privilege escalation by taking advantage of some known null de-reference exploit, typically in the kernel.</td>
</tr>
<tr>
<td>memory_exclusion_list</td>
<td>The executable files to exclude from Memory Protection. This must be a relative path to the excluded executable file.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>&quot;memory_exclusion_list&quot;: [</td>
</tr>
<tr>
<td></td>
<td>&quot;\temp&quot;</td>
</tr>
<tr>
<td></td>
<td>]</td>
</tr>
<tr>
<td>policy</td>
<td>Various policy settings are contained within this section. Some policy settings are enabled under policy and configured in a different section, like device_control and logpolicy.</td>
</tr>
<tr>
<td></td>
<td>■ auto_blocking: Enables or disables the Auto Quarantine setting for Unsafe and Abnormal files.</td>
</tr>
<tr>
<td></td>
<td>Note: filetype_actions must also be set for Unsafe (threat_files) and Abnormal (suspicious_files) files.</td>
</tr>
<tr>
<td></td>
<td>• 0 - Disabled</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| auto_delete:        | Setting to automatically delete quarantined files after a set number of days.  
| enabled             | Note: If this feature is enabled, set `days_until_deleted` for the number of days to retain a quarantined file.  
|                     | - 0 - Disabled  
|                     | - 1 - Enabled  
| auto_uploading:     | Setting to automatically upload files that Cylance has not seen before. Cylance will perform an analysis on the file and provide details to assist in manual analysis and triage.  
| enabled             | Note: `filetype_actions` must also be set for Unsafe (`threat_files`) and Abnormal (`suspicious_files`) files for Auto-Upload.  
|                     | - 0 - Disabled  
|                     | - 1 - Enabled  
| autoit_auto_uploading: |  
| custom_thumbprint:  |  
| days_until_deleted:| Setting for the number of days to retain a quarantined file. Quarantined files older than the set number of days will be automatically deleted. The minimum number of days is 14, the maximum number of days is 365.  
| enabled             | Note: To use the `days_until_deleted` feature, the `auto_delete` setting must be enabled.  
|                     | Example:  
|                     | {  
|                     |     "name": "days_until_deleted",  
|                     |     "value": "14"  
|                     | }  
| device_control:    | Setting to enable or disable the Device Control feature. Use `device_control` settings to Block or give FullAccess to specific types of USB mass storage devices. See `device_control` towards the beginning of this table.  
| enabled             | - 0 - Disabled  
|                     | - 1 - Enabled  
| docx_auto_uploading: |  
| full_disc_scan:     | Setting to have Cylance analyze all executable files on disk to detect any dormant threats. This is the Background Threat Detection (BTD) setting.  
| enabled             | - 0 - Disabled  
|                     | - 1 - Run Recurring (performs a scan every nine days)  
|                     | - 2 - Run Once (runs a full disk scan upon installation only)  
| kill_running_threats: | Setting to kill processes and child processes  

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>logpolicy:</td>
<td>This setting is not used.</td>
</tr>
<tr>
<td>low_confidence_threshold:</td>
<td>Setting to adjust the Cylance Score threshold between Unsafe and Abnormal files. The default is -600, therefore:</td>
</tr>
<tr>
<td></td>
<td>* A Cylance Score of -600 to -1000 is Unsafe.</td>
</tr>
<tr>
<td></td>
<td>* A Cylance Score of 0 to -599 is Abnormal.</td>
</tr>
<tr>
<td></td>
<td>* A Cylance Score greater than 0 is Safe.</td>
</tr>
<tr>
<td>Example:</td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;name&quot;: &quot;low_confidence_threshold&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;value&quot;: &quot;-600&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>memory_exploit_detection:</td>
<td>Setting to enable or disable the Memory Protection feature.</td>
</tr>
<tr>
<td>Note:</td>
<td>Also set the <code>memoryviolations</code> (memory_violations, memory_violations_ext, and memory_exclusion_list).</td>
</tr>
<tr>
<td></td>
<td>* 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>* 1 - Enabled</td>
</tr>
<tr>
<td>ole_auto_uploading:</td>
<td></td>
</tr>
<tr>
<td>optics:</td>
<td>Setting to enable or disable CylanceOPTICS.</td>
</tr>
<tr>
<td>Note:</td>
<td>Also set the other CylanceOPTICS (optics_) settings.</td>
</tr>
<tr>
<td></td>
<td>* 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>* 1 - Enabled</td>
</tr>
<tr>
<td>optics_application_control_auto_upload:</td>
<td>Setting to allow the automatic uploading of Application Control related Focus Data.</td>
</tr>
<tr>
<td></td>
<td>* 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>* 1 - Enabled</td>
</tr>
<tr>
<td>optics_malware_auto_upload:</td>
<td>Setting to allow the automatic uploading of Threat related Focus Data.</td>
</tr>
<tr>
<td></td>
<td>* 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>* 1 - Enabled</td>
</tr>
<tr>
<td>optics_memory_defense_auto_upload:</td>
<td>Setting to allow the automatic uploading of Memory Protection related Focus Data.</td>
</tr>
<tr>
<td></td>
<td>* 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>* 1 - Enabled</td>
</tr>
<tr>
<td>optics_script_control_auto_upload:</td>
<td>Setting to allow the automatic uploading of Script Control related Focus Data.</td>
</tr>
<tr>
<td></td>
<td>* 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>* 1 - Enabled</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>optics_sensors_advanced_executable_parsing</td>
<td>Setting to enable recording data fields associated with Portable Executable (PE) files, such as File version, Import functions, and Packer types. This is Enhanced Portable Executable Parsing in the policy settings.</td>
</tr>
<tr>
<td></td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>optics_sensors_advanced_powershell_visibility</td>
<td>Setting to enable recording commands, arguments, scripts, and content entered directly into the PowerShell Console and the Powershell Integrated Scripting Environment (ISE).</td>
</tr>
<tr>
<td></td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>optics_sensors_advanced_wmi_visibility</td>
<td>Setting to enable recording additional Windows Management Instrumentation (WMI) attributes and parameters.</td>
</tr>
<tr>
<td></td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Disabled</td>
</tr>
<tr>
<td>optics_sensors_dns_visibility</td>
<td>Setting to enable recording commands and arguments of commands issued directly or indirectly to the Windows Management Instrumentation (WMI) interpreter.</td>
</tr>
<tr>
<td></td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>optics_sensors_enhanced_process_hooking_visibility</td>
<td>Setting to enable recording process information from the Win32 API and Kernel Audit messages to detect forms of process hooking and injection.</td>
</tr>
<tr>
<td></td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>optics_sensors_private_network_address_visibility</td>
<td>Setting to enable recording network connections within the RFC 1918 and RFC 3419 address spaces.</td>
</tr>
<tr>
<td></td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>optics_sensors_windows_event_log_visibility</td>
<td>Setting to enable recording Windows Security Events and their associated attributes.</td>
</tr>
<tr>
<td></td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>optics_set_disk_usage_maximum_fixed</td>
<td>Setting the maximum amount of device storage reserved for use by CylanceOPTICS, in MB. The minimum value is 500 and the maximum value is 1000.</td>
</tr>
</tbody>
</table>

Example:
```
{
    "name": "optics_set_disk_usage_maximum_fixed",
    "value": "1000"
}
```
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>optics_show_notifications</td>
<td>Setting to enable or disable Desktop Notifications on the endpoint for CylanceOPTICS events.</td>
</tr>
<tr>
<td>• 0 - Disabled</td>
<td></td>
</tr>
<tr>
<td>• 1 - Enabled</td>
<td></td>
</tr>
<tr>
<td>pdf_auto_uploading</td>
<td></td>
</tr>
<tr>
<td>powershell_auto_uploading</td>
<td></td>
</tr>
<tr>
<td>prevent_service_shutdown</td>
<td>Setting that protects the Cylance service from being shutdown, either manually or by another process.</td>
</tr>
<tr>
<td>• 0 - Disabled</td>
<td></td>
</tr>
<tr>
<td>• 1 - Enabled</td>
<td></td>
</tr>
<tr>
<td>python_auto_uploading</td>
<td></td>
</tr>
<tr>
<td>sample_copy_path</td>
<td>Setting to copy all file samples to a network share (CIFS/SMB). Example:</td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>&quot;name&quot;: &quot;sample_copy_path&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;value&quot;: &quot;\server_name\shared_folder&quot;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>scan_exception_list</td>
<td>Setting to exclude specific folders and subfolders from being scanned by full_disc_scan and watch_for_new_files. Set the value to the absolute path for the excluded files. Example:</td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>&quot;name&quot;: &quot;scan_exception_list&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;value&quot;: [</td>
<td></td>
</tr>
<tr>
<td>&quot;c:\temp&quot;</td>
<td></td>
</tr>
<tr>
<td>]</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>scan_max_archive_size</td>
<td>Setting for the maximum archive file size (in MB) to be scanned. The value can be 0 to 150. If set to 0, then archive files will not be scanned. Example:</td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>&quot;name&quot;: &quot;scan_max_archive_size&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;value&quot;: &quot;0&quot;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>script_control</td>
<td>Setting to enable or disable the Script Control feature. Note: Also set the script_control settings (see below in this table).</td>
</tr>
<tr>
<td>• 0 - Disabled</td>
<td></td>
</tr>
<tr>
<td>• 1 - Enabled</td>
<td></td>
</tr>
<tr>
<td>show_notifications</td>
<td>Setting to enable or disable Desktop Notifications on the endpoint for CylancePROTECT events.</td>
</tr>
<tr>
<td>• 0 - Disabled</td>
<td></td>
</tr>
<tr>
<td>• 1 - Enabled</td>
<td></td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>threat_report_limit:</td>
<td>The number of threats to upload to the Console.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>{ &quot;name&quot;: &quot;threat_report_limit&quot;, &quot;value&quot;: &quot;500&quot; }</code></td>
</tr>
<tr>
<td>trust_files_in_scan_exception_list:</td>
<td>Setting to Allow Execution of files in the excluded folders. This is related to the scan_exception_list.</td>
</tr>
<tr>
<td>0 - Disabled</td>
<td>1 - Enabled (Allows execution of files in the excluded folders)</td>
</tr>
<tr>
<td>watch_for_new_files:</td>
<td>Setting to analyze new or modified executable files for threats.</td>
</tr>
<tr>
<td>0 - Disabled</td>
<td>1 - Enabled</td>
</tr>
<tr>
<td>optics_set_disk_usage_maximum_fixed:</td>
<td>Setting the maximum amount of device storage reserved for use by CylanceOPTICS, in MB. The minimum value is 500 and the maximum value is 1000.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>{ &quot;name&quot;: &quot;optics_set_disk_usage_maximum_fixed&quot;, &quot;value&quot;: &quot;1000&quot; }</code></td>
</tr>
<tr>
<td>optics_show_notifications:</td>
<td>Setting to enable or disable Desktop Notifications on the endpoint for CylanceOPTICS events.</td>
</tr>
<tr>
<td>0 - Disabled</td>
<td>1 - Enabled</td>
</tr>
<tr>
<td>pdf_auto_uploading:</td>
<td></td>
</tr>
<tr>
<td>powershell_auto_uploading:</td>
<td></td>
</tr>
<tr>
<td>prevent_service_shutdown:</td>
<td>Setting that protects the Cylance service from being shutdown, either manually or by another process.</td>
</tr>
<tr>
<td>0 - Disabled</td>
<td>1 - Enabled</td>
</tr>
<tr>
<td>python_auto_uploading:</td>
<td></td>
</tr>
<tr>
<td>sample_copy_path:</td>
<td>Setting to copy all file samples to a network share (CIFS/SMB).</td>
</tr>
<tr>
<td>Example:</td>
<td><code>{ &quot;name&quot;: &quot;sample_copy_path&quot;, &quot;value&quot;: &quot;\\server_name\shared_folder&quot; }</code></td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>scan_exception_list: Setting to exclude specific folders and subfolders from being scanned by full_disc_scan and watch_for_new_files. Set the value to the absolute path for the excluded files.</td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;name&quot;: &quot;scan_exception_list&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;value&quot;: [</td>
</tr>
<tr>
<td></td>
<td>&quot;c:\temp&quot;</td>
</tr>
<tr>
<td></td>
<td>]</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>scan_max_archive_size: Setting for the maximum archive file size (in MB) to be scanned. The value can be 0 to 150. If set to 0, then archive files will not be scanned.</td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;name&quot;: &quot;scan_max_archive_size&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;value&quot;: &quot;0&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>script_control: Setting to enable or disable the Script Control feature.</td>
<td><strong>Note:</strong> Also set the script_control settings (see below in this table).</td>
</tr>
<tr>
<td></td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>show_notifications: Setting to enable or disable Desktop Notifications on the endpoint for CylancePROTECT events.</td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>threat_report_limit: The number of threats to upload to the Console.</td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>{</td>
</tr>
<tr>
<td></td>
<td>&quot;name&quot;: &quot;threat_report_limit&quot;,</td>
</tr>
<tr>
<td></td>
<td>&quot;value&quot;: &quot;500&quot;</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>trust_files_in_scan_exception_list: Setting to Allow Execution of files in the excluded folders. This is related to the scan_exception_list.</td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled (Allows execution of files in the excluded folders)</td>
</tr>
<tr>
<td>watch_for_new_files: Setting to analyze new or modified executable files for threats.</td>
<td>- 0 - Disabled</td>
</tr>
<tr>
<td></td>
<td>- 1 - Enabled</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>policy_id</td>
<td>The unique identifier for the policy.</td>
</tr>
<tr>
<td>policy_name</td>
<td>The name of the policy. The name must be unique to your tenant.</td>
</tr>
</tbody>
</table>
| script_control | The policy settings for Script Control.  
  *script_control* must be enabled (set to "1") under *policy*. |
| activescript_settings |  
  - control_mode: Settings for Active Script.  
    - Alert: An alert is sent when an Active Script event occurs. The Active Script is allowed to run.  
    - Block: The Active Script is blocked and an alert is sent. |
| global_settings |  
  - allowed_folders: The relative path to scripts that are allowed to run when Script Control is enabled. Script Control Folder Exclusions apply to all Agent versions (Agent 1310 or higher).  
    - Example:  
      ```json  
      "allowed_folders": [  
        "\temp_scriptcontrol"  
      ]  
      ```  
  - control_mode: Setting to enable or disable Script Control for Agent version 1370 or lower. This works for Active Scripts and PowerShell. This does not work for Macros. To use Script Control with Macros, use Agent version 1380 or higher.  
    - Allow: An alert is sent when an Active Script or PowerShell event occurs. The script is allowed to run.  
    - Block: The Active Script or PowerShell is blocked and an alert is sent. |
| macro_settings |  
  - control_mode: Settings for Microsoft Office Macros.  
    - Alert: An alert is sent when an Office Macro event occurs. The Macro is allowed to run.  
    - Block: The Office Macro is blocked and an alert is sent. |
| powershell_settings |  
  - console_mode: The PowerShell Console is blocked to prevent PowerShell command usage, including one-liners. To use this feature, the PowerShell *control_mode* must be set to Block.  
  - control_mode:  
    - Alert: An alert is sent when a PowerShell script event occurs. The PowerShell script is allowed to run.  
    - Block: The PowerShell script is blocked and an alert is sent. |

**About disabling Script Control**  
For Agent versions 1430 and higher, you can disable Script Control for Active
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>script_control</td>
<td>Script, PowerShell, or Macros. Disabling Script Control allows the selected script type to run and does not send an alert to the Console. To disable Script Control for a specific script type, do not include the script type in the Create Policy API request. Example: Script Control for Macros is disabled.</td>
</tr>
<tr>
<td>user_id</td>
<td>The unique ID for the user creating the policy. Only administrators can create policies. Note: To get the user_id, use &quot;Get Users on page 31.&quot;</td>
</tr>
</tbody>
</table>

**Delete Policy**

Delete a policy from a tenant.

**Service Endpoint:**

- /policies/v2/{tenant_policy_id}

**Example with user_id:** https://protectapi.cylance.com/policies/v2/d5c6d6a3-0599-4fb5-96bc-0fdc7eacb6ea

**Method:**

- HTTP/1.1 DELETE

**Request Headers:**

- Authorization: Bearer <JWT Token returned by Auth API> with the policy:delete scope encoded

**Request:**

None
Response:

204 No Content - Policy deleted

Delete Policies

Delete multiple polices from a tenant.

Service Endpoint:

- /policies/v2

Example with user_id: https://protectapi.cylance.com/policies/v2

Method:

- HTTP/1.1 DELETE

Request Headers:

- Authorization: Bearer <JWT Token returned by Auth API> with the policy:delete scope encoded
- Content-Type: application/json

Request:

```
{
    "tenant_policy_ids": [
        "d5c6d6a3-0599-4fb5-96bc-0fdc7eacb6ea",
        "376e21d1-f227-49c4-85fb-d9be1e5d766b",
        "b7a4a177-e385-489b-bcb0-3a4f25276320"
    ]
}
```

Response:

204 No Content - Policy deleted
Zone API
Create Zone

Add a zone to a tenant.

Service Endpoint:

- /zones/v2

Example: https://protectapi.cylance.com/zones/v2

Method:

- HTTP/1.1 POST

Request Header:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the zone:create scope encoded.
- Content-Type: application/json

Request:

```json
{
   "name": "Test Zone",
   "policy_id": "d5c6d6a3-0599-4fb5-96bc-0fdc7eacb6ea",
   "criticality": "Normal"
}
```

Note: The policy_id or criticality requests can be removed if they are not needed. If the policy_id is removed, the zone is created with the Default policy. If the criticality is removed, the zone is created with the Normal criticality.

Response:

201 Created

```json
{
   "id": "d27ff5c4-5c0d-4f56-a00d-a1fb297e440e",
   "name": "Test Zone",
   "criticality": "Normal",
   "policy_id": "d5c6d6a3-0599-4fb5-96bc-0fdc7eacb6ea",
   "date_created": "2019-03-04T23:03:44",
}
```

400 Bad Request - The Tenant ID could not be retrieved from the JWT token specified in the Authorization header.
- The zone name value provided is not a valid value for a name.
- The zone criticality provided is not a valid value for criticality.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
409 Conflict - The zone name provided already exists in your organization.
500 Internal Server Error - An unforeseeable error has occurred.

**Request JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>criticality</td>
<td>The value of the zone - Low, Normal, or High. If a value is not included, then the default (Normal) is assigned.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the Zone.</td>
</tr>
<tr>
<td>policy_id</td>
<td>The unique ID for the policy assigned to the Zone. If a policy_id is not included, the Default policy is assigned.</td>
</tr>
</tbody>
</table>

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>criticality</td>
<td>The value of the zone - Low, Normal, or High.</td>
</tr>
<tr>
<td>date_created</td>
<td>The date and time (in UTC) when the Zone was created.</td>
</tr>
<tr>
<td>id</td>
<td>The unique ID for the Zone.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the Zone.</td>
</tr>
<tr>
<td>policy_id</td>
<td>The unique ID for the policy assigned to the Zone.</td>
</tr>
</tbody>
</table>

**Get Zones**

Request a page with a list of Zones resources belonging to a tenant, sorted by the created date, in descending order (most recent user registered listed first). The page number and page size parameters are optional. When the values are not specified, the default values are 1 and 10 respectively. The maximum page size that can be specified is 200 entries per page.

**Service Endpoint:**

- /zones/v2?page=m&page_size=n

Append the following optional query string parameters:
- page: The page number to request
- page_size: The number of device records to retrieve per page

For example, to return the first page with 100 users:
https://protectapi.cylance.com/zones/v2?page=1&page_size=100

Method:
- HTTP/1.1 GET

Request Headers:
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the zone:list scope encoded

Request:
None

Response:
200 OK

```json
{
  "page_number": "1",
  "page_size": "10",
  "total_pages": "1",
  "total_number_of_items": "1",
  "page_items": [
    {
      "id": "d27ff5c4-5c0d-4f56-a00d-a1fb297e440e",
      "name": "Test Zone",
      "criticality": "Normal",
      "zone_rule_id": null,
      "policy_id": "d5c6d6a3-0599-4fb5-96bc-0f6c7eacb6ea",
      "update_type_in": "Production",
      "date_created": "2019-03-04T23:03:44",
      "date_modified": "2019-03-04T23:03:44"
    }
  ]
}
```

400 Bad Request - Returned for the following reasons:
- The Tenant ID cannot be retrieved from the JWT token
- The zone unique identifier is not valid.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
404 Not Found - The zone requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>criticality</td>
<td>The value of the zone - Low, Normal, or High.</td>
</tr>
<tr>
<td>date_created</td>
<td>The date and time (in UTC) when the zone was created.</td>
</tr>
<tr>
<td>date_modified</td>
<td>The date and time (in UTC) when the zone was last modified.</td>
</tr>
<tr>
<td>id</td>
<td>The unique ID for the zone.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the zone.</td>
</tr>
<tr>
<td>policy_id</td>
<td>The unique ID for the policy assigned to the zone.</td>
</tr>
<tr>
<td>update_type</td>
<td>The update type for the zone - Production, Pilot, or Test.</td>
</tr>
<tr>
<td>zone_rule_id</td>
<td>The unique ID for the zone rule created for the zone. If there is no zone rule, then null is displayed.</td>
</tr>
</tbody>
</table>

Get Zone

Request zone information for a specific zone in a tenant.

Service Endpoint:

- /zones/v2/{unique_zone_id}

For example, to return the first page with 100 users:
https://protectapi.cylance.com/zones/v2/d27ff5c4-5c0d-4f56-a00d-a1fb297e440e

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the zone:read scope encoded

Request:
None

Response:

200 OK

```json
{
    "id": "d27ff5c4-5c0d-4f56-a00d-a1fb297e440e",
    "name": "Test Zone",
    "criticality": "Normal",
    "zone_rule_id": null,
    "policy_id": "d5c6d6a3-0599-4fb5-96bc-0fdc7eacb6ea",
    "update_type_in": "Production",
    "date_created": "2019-03-04T23:03:44",
    "date_modified": "2019-03-04T23:03:44"
}
```

400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The zone unique identifier is not valid.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The zone requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>criticality</td>
<td>The value of the zone - Low, Normal, or High.</td>
</tr>
<tr>
<td>date_created</td>
<td>The date and time (in UTC) when the zone was created.</td>
</tr>
<tr>
<td>date_modified</td>
<td>The date and time (in UTC) when the zone was last modified.</td>
</tr>
<tr>
<td>id</td>
<td>The unique ID for the zone.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the zone.</td>
</tr>
<tr>
<td>policy_id</td>
<td>The unique ID for the policy assigned to the zone.</td>
</tr>
<tr>
<td>update_type</td>
<td>The update type for the zone - Production, Pilot, or Test.</td>
</tr>
<tr>
<td>zone_rule_id</td>
<td>The unique ID for the zone rule created for the zone. If there is no zone rule, then null is displayed.</td>
</tr>
</tbody>
</table>
Get Device Zones

Request a page with a list of Zone resources for a specified device, belonging to a tenant, sorted by the created date, in descending order (most recent user registered listed first). The page number and page size parameters are optional. When the values are not specified, the default values are 1 and 10 respectively. The maximum page size that can be specified is 200 entries per page.

**Service Endpoint:**

- `/zones/v2/{unique_device_id}/zones?page=m&page_size=n`

Append the following optional query string parameters:

- page: The page number to request
- page_size: The number of device records to retrieve per page

For example, to return the first page with 100 users:

```plaintext
https://protectapi.cylance.com/zones/v2/e378dacb-9324-453a-b8c6-5a8406952195/zones?page=1&page_size=100
```

**Method:**

- `HTTP/1.1 GET`

**Request Headers:**

- `Accept: application/json`
- `Authorization: Bearer <JWT Token returned by Auth API> with the zone:list scope encoded`

**Request:**

None

**Response:**

200 OK

```json
{
  "page_number": "1",
  "page_size": "10",
  "total_pages": "1",
  "total_number_of_items": "1",
  "page_items": [
    {
      "id": "d27ff5c4-5c0d-4f56-a00d-a1fb297e440e",
      "name": "Test Zone",
      "criticality": "Normal",
      "zone_rule_id": null,
    }
  ]
}
```
400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The page number or page size specified is less than or equal to zero, or the page size is greater than 200

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

500 Internal Server Error - An unforeseeable error has occurred.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>criticality</td>
<td>The value of the zone - Low, Normal, or High.</td>
</tr>
<tr>
<td>date_created</td>
<td>The date and time (in UTC) when the zone was created.</td>
</tr>
<tr>
<td>date_modified</td>
<td>The date and time (in UTC) when the zone was last modified.</td>
</tr>
<tr>
<td>id</td>
<td>The unique ID for the zone.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the zone.</td>
</tr>
<tr>
<td>policy_id</td>
<td>The unique ID for the policy assigned to the zone.</td>
</tr>
<tr>
<td>update_type</td>
<td>The update type for the zone - Production, Pilot, or Test.</td>
</tr>
<tr>
<td>zone_rule_id</td>
<td>The unique ID for the zone rule created for the zone. If there is no zone rule, then null is displayed.</td>
</tr>
</tbody>
</table>

Update Zone

Update a zone in a tenant.

Service Endpoint:

- /zones/v2/{unique_zone_id}
For example, to return the first page with 100 users:
https://protectapi.cylance.com/zones/v2/d27ff5c4-5c0d-4f56-a00d-a1fb297e440e

Method:
- HTTP/1.1 PUT

Request Headers:
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the zone:update scope encoded

Request:
```json
{
    "name": "Test Policy",
    "policy_id": "d5c6d6a3-0599-4fb5-96bc-0f8c7eacb6ea",
    "criticality": "Normal"
}
```

Response:
200 OK

400 Bad Request - Returned for the following reasons:
- The Tenant ID cannot be retrieved from the JWT token
- The zone name provided is not a valid value for a name
- The zone criticality provided is not a valid value for criticality

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The zone requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>criticality</td>
<td>The value of the zone - Low, Normal, or High.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the zone.</td>
</tr>
<tr>
<td>policy_id</td>
<td>The unique ID for the policy assigned to the zone.</td>
</tr>
</tbody>
</table>
Delete Zone

Delete (remove) a zone in a tenant.

Service Endpoint:

- /zones/v2/{unique_zone_id}

For example, to return the first page with 100 users:
https://protectapi.cylance.com/zones/v2/d27ff5c4-5c0d-4f56-a00d-a1fb297e440

Method:

- HTTP/1.1 DELETE

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the zone:delete scope encoded

Request:

None

Response:

200 OK

400 Bad Request - The Tenant ID could not be retrieved from the JWT token specified in the Authorization header.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The zone requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.
Threat API
Get Threats

Request a page with a list of threat resources belonging to a tenant, sorted by the last found date, in descending order (most recent found threat listed first). The page number and page size parameters are optional. When the values are not specified, the default values are 1 and 10 respectively. The maximum page size that can be specified is 200 entries per page.

Service Endpoint:

```
/threats/v2?page=m&page_size=n
```

Append the following optional query string parameters:

- `page`: The page number to request
- `page_size`: The number of device records to retrieve per page
- `start_time`: The start of the time range. Format is YYYY-MM-DDThh:mm:ss.SSSZ (ISO 8601 date/time format). Required if using an `end_time`.
- `end_time`: The end of the time range. Format is: YYYY-MM-DDThh:mm:ss.SSSZ (ISO 8601 date/time format). Optional. The default value is now.

For example, to return the first page with 100 threats, using a time range:

```
https://protectapi.cylance.com/threats/v2?page=1&page_size=100&start_time=2019-11-01T12:00:00.000Z&end_time:2019-11-30T12:00:00.000Z
```

Method:

- HTTP/1.1 GET

Request Headers:

- `Accept`: application/json
- `Authorization Bearer <JWT Token returned by Auth API>` with the `threat:list` scope encoded

Request:

None

Response:

```
200 OK
{
    "page_number": "1",
    "page_size": "10",
    "total_pages": "1",
    "total_number_of_items": "1",
    "page_items": [ ...
```
400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The page number or page size specified is less than or equal to zero, or the page size is greater than 200
- The start time cannot be less than 01-01-1753 0:00.
- The start time should always be before the end time. The service will not automatically translate to a valid time range.
- The times are in the future.
- The query has an end time but does not have a start time.
- The date/time parameters are invalid.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The device resources page requested doesn't exist.

429 Too Many Requests - The rate limit has been reached.

500 Internal Server Error - An unforeseeable error has occurred.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>av_industry</td>
<td>The score provided by the antivirus industry. If there is no antivirus industry score, then null is displayed.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>classification</td>
<td>The threat classification for the threat. See &quot;Threat Classifications&quot; on page 327 for more information.</td>
</tr>
<tr>
<td>cylance_score</td>
<td>The Cylance Score assigned to the threat. The Cylance API returns a raw score of -1 to 1. Threats have a negative raw score, while safe files have a positive raw score. The Cylance Console only displays threats and uses a score of 1 to 100. A raw score of -1 equals a Console score of 100.</td>
</tr>
<tr>
<td>file_size</td>
<td>The size of the file, in bytes.</td>
</tr>
<tr>
<td>global_quarantine</td>
<td>Identifies if the threat is on the Global Quarantine list.</td>
</tr>
<tr>
<td></td>
<td>- false - The file is not on the Global Quarantine list.</td>
</tr>
<tr>
<td></td>
<td>- true - The file is on the Global Quarantine list.</td>
</tr>
<tr>
<td>last_found</td>
<td>The date and time the threat was last found on any device in your organization.</td>
</tr>
<tr>
<td>md5</td>
<td>The MD5 hash for the threat.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the threat.</td>
</tr>
<tr>
<td>page_number</td>
<td>The page number requested.</td>
</tr>
<tr>
<td>page_size</td>
<td>The page size requested.</td>
</tr>
<tr>
<td>safelisted</td>
<td>Identifies if the threat is on the Safe List.</td>
</tr>
<tr>
<td></td>
<td>- false - The file is not on the Safe List.</td>
</tr>
<tr>
<td></td>
<td>- true - The file is on the Safe List.</td>
</tr>
<tr>
<td>sha256</td>
<td>The SHA256 hash for the threat.</td>
</tr>
<tr>
<td>sub_classification</td>
<td>The threat sub-classification for the threat.</td>
</tr>
<tr>
<td>total_number_of_items</td>
<td>The total number of resources.</td>
</tr>
<tr>
<td>total_pages</td>
<td>The total number of pages that can be retrieved, based on the page size specified.</td>
</tr>
<tr>
<td>unique_to_cylance</td>
<td>The threat is identified by Cylance but not by other antivirus sources.</td>
</tr>
<tr>
<td></td>
<td>- false - The file has been identified by other antivirus sources.</td>
</tr>
<tr>
<td></td>
<td>- true - The file has only been identified as a threat by Cylance.</td>
</tr>
</tbody>
</table>
Get Threat

Request threat details for a specific threat.

Service Endpoint:

- /threats/v2/{threat_sha256}

Example:
https://protectapi.cylance.com/threats/v2/bf17366ee3bb8068a9ad70fc9e68496e7e311a055bf4fffeff53cc5d29ccce52

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the threat:read scope encoded

Request:

None

Response:

200 OK

{
   "name": "threatfile.exe",
   "sha256": "bf17366ee3bb8068a9ad70fc9e68496e7e311a055bf4fffeff53cc5d29ccce52",
   "md5": "d41d8cd98f00b204e9800998ecf8427e",
   "signed": true,
   "cylance_score": -1,
   "av_industry": null,
   "classification": "PUP",
   "sub_classification": "Generic",
   "global_quarantine": false,
   "safelisted": false,
   "cert_publisher": "Publisher Name",
   "cert_issuer": "Issuer Name",
   "cert_timestamp": "0001-01-01T00:00:00",
   "file_size": 1500000,
   "unique_to_cylance": false,
   "running": false,
   "auto_run": false,
   "detected_by": "Background Threat Detection"
}
400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The threat hash ID specified is invalid

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The threat requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| auto_run                 | Indicates if the file is set to automatically run on system startup.  
  - false - The file is not set to automatically run on system startup.  
  - true - The file is to automatically run on system startup. |
| av_industry              | The score provided by the antivirus industry. If there is no antivirus industry score, then null is displayed.                               |
| cert_issuer              | The ID for the certificate issuer.                                                                                                         |
| cert_publisher           | The ID for the certificate publisher.                                                                                                       |
| cert_timestamp           | The date and time (in UTC) when the file was signed using the certificate.                                                                   |
| classification           | The threat classification for the threat. See "Threat Classifications" on page 327 for more information.                                    |
| cylance_score            | The Cylance Score assigned to the threat.  
  The Cylance API returns a raw score of -1 to 1. Threats have a negative raw score, while safe files have a positive raw score.  
  The Cylance Console only displays threats and uses a score of 1 to 100. A raw score of -1 equals a Console score of 100. |
| detected_by              | The name of the [[[Undefined variable General.Company_Name]]] module that detected the threat.                                                |
| file_size                | The size of the file, in bytes.                                                                                                             |
| global_quarantine        | Identifies if the threat is on the Global Quarantine list.  
  - false - The file is not on the Global Quarantine list.  
  - true - The file is on the Global Quarantine list. |
<p>| md5                      | The MD5 hash for the threat.                                                                                                               |</p>
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the threat.</td>
</tr>
<tr>
<td>running</td>
<td>Identifies if the threat is executing, or another executable loaded or called it.</td>
</tr>
<tr>
<td>safelisted</td>
<td>Identifies if the threat is on the Safe List.</td>
</tr>
<tr>
<td>sha256</td>
<td>The SHA256 hash for the threat.</td>
</tr>
<tr>
<td>signed</td>
<td>Identifies if the file is signed or not signed.</td>
</tr>
<tr>
<td>sub_classification</td>
<td>The threat sub-classification for the threat. See &quot;Threat Classifications&quot; on page 327 for more information.</td>
</tr>
<tr>
<td>unique_to_cylance</td>
<td>The threat was identified by Cylance but not by other antivirus sources.</td>
</tr>
</tbody>
</table>

### Get Threat Devices

Request a list of devices affected by a specific threat.

**Note:** Only one file_path is listed per page_item, therefore the same device could have multiple entries, one entry per file_path.

**Service Endpoint:**

- /threats/v2/(threat_sha256)/devices?page=m&page_size=n

Append the following optional query string parameters:

- page: The page number to request
- page_size: The number of device records to retrieve per page
- threat_sha256: The SHA256 hash for the threat.

For example, to return the first page with 100 devices that have the specified threat:

https://protectapi.cylance.com/threats/v2/bf17366ee3bb8068a9ad70fc9e68496e7e311a055bf4ffeef53cc5d29ccee52/devices?page1&page_size=100

**Method:**
HTTP/1.1 GET

Request Headers:
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the threat:devicelist scope encoded

Request:
None

Response:
200 OK

```json
{
    "page_number": "1",
    "page_size": "10",
    "total_pages": "1",
    "total_number_of_items": "1",
    "page_items": [
        {
            "id": "e378dacb-9324-453a-b8c6-5a8406952195",
            "name": "User-Laptop-A123",
            "state": "Online",
            "agent_version": "2.0.1530",
            "policy_id": "d5c666a3-0599-4fb5-96bc-0fde7eac6eae",
            "date_found": "2019-05-14T19:54:26",
            "file_status": "Quarantined",
            "file_path": "C:\temp\threatfile.exe",
            "ip_addresses": [
                "123.45.67.89"
            ],
            "mac_addresses": [
                "00-00-00-00-00-00"
            ]
        }
    ]
}
```

400 Bad Request - Returned for the following reasons:
- The Tenant ID cannot be retrieved from the JWT token
- The threat SHA256 hash is invalid
- The page number or page size specified is less than or equal to zero, or the page size is greater than 200

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
500 Internal Server Error - An unforeseeable error has occurred.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent_version</td>
<td>The CylancePROTECT Agent version installed on the device.</td>
</tr>
<tr>
<td>date_found</td>
<td>The date and time (in UTC) when the threat was found on the device.</td>
</tr>
<tr>
<td>file_path</td>
<td>The path where the file was found on the device.</td>
</tr>
<tr>
<td>file_status</td>
<td>Current quarantine status of the file on the device.</td>
</tr>
<tr>
<td>id</td>
<td>The endpoint's unique identifier.</td>
</tr>
<tr>
<td>ip_addresses</td>
<td>The list of IP addresses for the device.</td>
</tr>
<tr>
<td>mac_addresses</td>
<td>The list of MAC addresses for the device.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the device.</td>
</tr>
<tr>
<td>page_number</td>
<td>The page number requested.</td>
</tr>
<tr>
<td>page_size</td>
<td>The page size requested.</td>
</tr>
<tr>
<td>policy_id</td>
<td>The unique identifier for the policy assigned to the device, or null if no policy is assigned.</td>
</tr>
<tr>
<td>state</td>
<td>The state of the device.</td>
</tr>
<tr>
<td>total_number_of_items</td>
<td>The total number of resources.</td>
</tr>
<tr>
<td>total_pages</td>
<td>The total number of pages that can be retrieved, based on the page size specified.</td>
</tr>
</tbody>
</table>
Get Threat Download URL

Request a download link for a given file. Use the download link to download the file.

Service Endpoint:

- /threats/v2/download/{threat_sha256}

Example:
https://protectapi.cylance.com/threats/v2/download/bf17366ee3bb8068a9ad70fc9e68496e7e311a055bf4ffeef53cc5d29ccce52

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the threat:read scope encoded

Request:

None

Response:

200 OK

```json
{
   "url": "https://cylance/download/url"
}
```

400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The threat SHA256 hash is invalid
  - The threat SHA256 hash is not associated with the tenant
  - The agent reported SHA1 and MD5 hashes that do not match values in the Console
  - SHA1 or MD5 hash is empty

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

500 Internal Server Error - An unforeseeable error has occurred.
### Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>The URL you can use to download the file. The API call only provides the URL, it does not download the file for you.</td>
</tr>
</tbody>
</table>
Get Memory Protection Events

Request a list of Memory Protection events.

Service Endpoint:

- `/memoryprotection/v2?page=m&page_size=n&start_time=t1&end_time=t2`

Append the following optional query string parameters:

- `page`: The page number to request
- `page_size`: The number of device records to retrieve per page
- `start_time`: The start of the time range. Format is YYYY-MM-DDThh:mm:ss.SSSZ (ISO 8601 date/time format). Required if using an `end_time`.
- `end_time`: The end of the time range. Format is: YYYY-MM-DDThh:mm:ss.SSSZ (ISO 8601 date/time format). Optional. The default value is now.
- `device_id`: Add a device ID to reduce the set of Memory Protection events. Default is null.

Example:
https://protectapi.cylance.com/memoryprotection/v2?page=1&page_size=100&start_time=2019-11-01T12:00:00&end_time:2019-11-30T12:00:00

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the memoryprotection:list scope encoded

Request:

None

Response:

200 OK

```json
{
    "page_number": 1,
    "page_size": 10,
    "total_pages": 1,
    "total_number_of_items": 1,
    "page_items": [
    
    ]
}
```
400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token.
- The Device Id parameter does not meet the format for an Id.
- The page number or page size specified is less than or equal to zero, or the page size is greater than 200.
- The time range between the start date and end date is larger than 90 days.
- The start time cannot be less than 01-01-1753 0:00.
- The start time should always be before the end time. The service will not automatically translate to a valid time range.
- The times are in the future.
- The query has an end time but does not have a start time.
- The date/time parameters are invalid.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The threat requested doesn't exist.

429 Too Many Requests - The rate limit has been reached.

500 Internal Server Error - An unforeseeable error has occurred.

**Response JSON Schema Descriptions**
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>The action take on the Memory Protection event.</td>
</tr>
<tr>
<td></td>
<td>0: None</td>
</tr>
<tr>
<td></td>
<td>1: Block</td>
</tr>
<tr>
<td></td>
<td>2: Terminate</td>
</tr>
<tr>
<td>agent_event_id</td>
<td>The unique identifier for the Memory Protection event, created by the Agent.</td>
</tr>
<tr>
<td>created</td>
<td>The date and time the Memory Protection event was created.</td>
</tr>
<tr>
<td>device_id</td>
<td>The unique identifier for the device.</td>
</tr>
<tr>
<td>device_image_file_event_id</td>
<td>The unique identifier for the Memory Protection event. Use this information for Get Memory Protection Event.</td>
</tr>
<tr>
<td>dll_version</td>
<td>The agent version that identified the Memory Protection event.</td>
</tr>
<tr>
<td>file_hash_id</td>
<td>The SHA256 hash for the threat.</td>
</tr>
<tr>
<td>file_version</td>
<td>The version number of the file that caused the Memory Protection event.</td>
</tr>
<tr>
<td>groups</td>
<td>The groups the user belongs to.</td>
</tr>
<tr>
<td>image_name</td>
<td>The path and name of the file that triggered the Memory Protection event.</td>
</tr>
<tr>
<td>process_id</td>
<td>The process ID of the Memory Protection event. This is generated by the operating system.</td>
</tr>
<tr>
<td>sid</td>
<td>The security identifier for the user, group, or other security principal. This is generated by the operating system.</td>
</tr>
<tr>
<td>username</td>
<td>The name of the user who was logged in to the device when the Memory Protection event occurred.</td>
</tr>
<tr>
<td>violation_type</td>
<td>The violation type number for the Memory Protection event. See &quot;Memory Violation Types&quot; on page 162 for more information.</td>
</tr>
</tbody>
</table>

### Get Memory Protection Event

Request details for a specific Memory Protection event.

**Service Endpoint:**

- `/memoryprotection/v2/{device_image_file_event_id}

**Example:** https://protectapi.cylance.com/memoryprotection/v2/40d04bf5-c5d7-495f-805a-28c6fc8ac12c

**Method:**
**HTTP/1.1 GET**

**Request Headers:**
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the memoryprotection:read scope encoded

**Request:**
None

**Response:**
200 OK

```json
{
   "device_image_file_event_id": "40d04bf5-c5d7-495f-805a-28c6fc8ac12c",
   "device_id": "string",
   "file_hash_id": "string",
   "image_name": "string",
   "process_id": 1234,
   "username": "TestUser",
   "groups": "BUILTIN\Users",
   "sid": "S-1-5-21-2825108384-3698480544-3979154530-1000",
   "violation_type": 1,
   "action": 0,
   "file_version": "1.7.0.0",
   "dll_version": "32-bit: 2.1.1550.15; 64-bit: 2.1.1550.15",
   "created": "2019-12-10T20:19:17",
   "agent_event_id": "14f85c31-e8b8-493b-a17e-4d6bade3d286"
}
```

400 No Tenant Id - The Tenant ID cannot be retrieved from the JWT token.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The threat requested doesn't exist.

429 Too Many Requests - The rate limit has been reached.

500 Internal Server Error - An unforeseeable error has occurred.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>The action take on the Memory Protection event.</td>
</tr>
<tr>
<td></td>
<td>- 0: None</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>agent_event_id</td>
<td>The unique identifier for the Memory Protection event, created by the Agent.</td>
</tr>
<tr>
<td>created</td>
<td>The date and time the Memory Protection event was created.</td>
</tr>
<tr>
<td>device_id</td>
<td>The unique identifier for the device.</td>
</tr>
<tr>
<td>device_image_file_event_id</td>
<td>The unique identifier for the Memory Protection event. Use this information for Get Memory Protection Event.</td>
</tr>
<tr>
<td>dll_version</td>
<td>The agent version that identified the Memory Protection event.</td>
</tr>
<tr>
<td>file_hash_id</td>
<td>The SHA256 hash for the threat.</td>
</tr>
<tr>
<td>file_version</td>
<td>The version number of the file that caused the Memory Protection event.</td>
</tr>
<tr>
<td>groups</td>
<td>The groups the user belongs to.</td>
</tr>
<tr>
<td>image_name</td>
<td>The path and name of the file that triggered the Memory Protection event.</td>
</tr>
<tr>
<td>process_id</td>
<td>The process ID of the Memory Protection event. This is generated by the operating system.</td>
</tr>
<tr>
<td>sid</td>
<td>The security identifier for the user, group, or other security principal. This is generated by the operating system.</td>
</tr>
<tr>
<td>username</td>
<td>The name of the user who was logged in to the device when the Memory Protection event occurred.</td>
</tr>
<tr>
<td>violation_type</td>
<td>The violation type number for the Memory Protection event. See <a href="#">&quot;Memory Violation Types&quot; below</a> for more information.</td>
</tr>
</tbody>
</table>

### Memory Violation Types

The following table provides a description of each Violation Type, the operating system on which the violation type is applied, and the violation type number returned by the Cylance API.

<table>
<thead>
<tr>
<th>#</th>
<th>Violation Type</th>
<th>Description</th>
<th>Applies To</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Stack Pivot</td>
<td>The stack for a thread has been replaced with a different stack. Generally the system will only allocate a single stack for a thread. An attacker would use a different stack to control execution in a way that is not blocked by Data Execution Prevention (DEP).</td>
<td>Windows, macOS, Linux</td>
</tr>
<tr>
<td>2</td>
<td>Stack Protect</td>
<td>The memory protection of a thread's stack has been modified to enable execution permissions. Stack memory should not be</td>
<td>Windows, macOS,</td>
</tr>
<tr>
<td>#</td>
<td>Violation Type</td>
<td>Description</td>
<td>Applies To</td>
</tr>
<tr>
<td>---</td>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>executable, so usually this means that an attacker is preparing to run malicious code stored in stack memory as part of an exploit, an attempt which would otherwise be blocked by Data Execution Prevention (DEP).</td>
<td>Linux</td>
</tr>
<tr>
<td>3</td>
<td>Overwrite Code</td>
<td>Code residing in a process's memory has been modified using a technique that may indicate an attempt to bypass Data Execution Prevention (DEP).</td>
<td>Windows</td>
</tr>
<tr>
<td>4</td>
<td>Remote Allocation of Memory</td>
<td>A process has allocated memory in another process. Most allocations will only occur within the same process. This generally indicates an attempt to inject code or data into another process, which may be a first step in reinforcing a malicious presence on a system.</td>
<td>macOS</td>
</tr>
<tr>
<td>5</td>
<td>Remote Mapping of Memory</td>
<td>A process has introduced code and/or data into another process. This may indicate an attempt to begin executing code in another process and thereby reinforce a malicious presence.</td>
<td>macOS</td>
</tr>
<tr>
<td>6</td>
<td>Remote Write to Memory</td>
<td>A process has modified memory in another process. This is usually an attempt to store code or data in previously allocated memory (see OutOfProcessAllocation) but it is possible that an attacker is trying to overwrite existing memory in order to divert execution for a malicious purpose.</td>
<td>Windows, macOS</td>
</tr>
<tr>
<td>7</td>
<td>Remote Write PE to Memory</td>
<td>A process has introduced code and/or data into another process. This may indicate an attempt to begin executing code in another process and thereby reinforce a malicious presence.</td>
<td>Windows</td>
</tr>
<tr>
<td>8</td>
<td>Remote Overwrite Code</td>
<td>A process has modified executable memory in another process. Under normal conditions executable memory will not be modified, especially by another process. This usually indicates an attempt to divert execution in another process.</td>
<td>Windows</td>
</tr>
<tr>
<td>9</td>
<td>Remote Unmap of Memory</td>
<td>A process has removed a Windows executable from the memory of another process. This may indicate an intent to replace the executable image with a modified copy for the purpose of diverting execution.</td>
<td>Windows</td>
</tr>
<tr>
<td>10</td>
<td>Remote Thread Creation</td>
<td>A process has created a new thread in another process. A process's threads are usually only created by that same process. This is generally used by an attacker to activate a malicious presence that has been injected into another process.</td>
<td>Windows, macOS</td>
</tr>
<tr>
<td>11</td>
<td>Remote APC Scheduled</td>
<td>A process has diverted the execution of another process's thread. This is generally used by an attacker to activate a malicious presence that has been injected into another process.</td>
<td>Windows</td>
</tr>
<tr>
<td>12</td>
<td>LSASS Read</td>
<td>Memory belonging to the Windows Local Security Authority process has been accessed in a manner that indicates an attempt to obtain users' passwords.</td>
<td>Windows</td>
</tr>
<tr>
<td>#</td>
<td>Violation Type</td>
<td>Description</td>
<td>Applies To</td>
</tr>
<tr>
<td>----</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>13</td>
<td>RAM Scraping</td>
<td>A process is trying to read valid magnetic stripe track data from another process. Typically related to point of sale systems (POS).</td>
<td>Windows</td>
</tr>
<tr>
<td>22</td>
<td>Zero Allocate</td>
<td>A null page has been allocated. The memory region is typically reserved, but in certain circumstances it can be allocated. Attacks can use this to setup privilege escalation by taking advantage of some known null de-reference exploit, typically in the kernel.</td>
<td>Windows, macOS</td>
</tr>
<tr>
<td>23</td>
<td>DYLD Injection</td>
<td>An environment variable has been set that will cause a shared library to be injected into a launched process. Attacks can modify the plist of applications like Safari or replace applications with bash scripts, that cause their modules to be loaded automatically when an application starts.</td>
<td>macOS, Linux</td>
</tr>
<tr>
<td>24</td>
<td>Malicious Payload</td>
<td>A generic shellcode and payload detection associated with exploitation has been detected.</td>
<td>Windows</td>
</tr>
</tbody>
</table>
Detection API
The CylanceOPTICS Detection API allows users to interact with Detection Events triggered by the CylanceOPTICS Context Analysis Engine (CAE). CAE allows users to take automated response actions against malicious or suspicious behavior detected on devices utilizing both machine learning models and static behavior-based rules.

The CylanceOPTICS Detection API enables further automation of analyzing, triaging, and responding to malicious or suspicious activity prevented or detected by CylanceOPTICS. The workflows currently available through this API include:

- Gathering a summary Detection Events that have occurred in a tenant including a Detection Event's ID, severity, description, occurrence time, associated device, and status.
- Gathering the specific Detection Details of Detection Events that have occurred in a tenant, including the artifacts associated with a Detection Event, the status of automated response actions that have been taken against a Detection Event, and other granular details that compose the Detection Event.
- Deleting a single or multiple Detection Events from a tenant.
- Updating a Detection Event's status and comments in a tenant.

**Get Detections**

Request a page with a list of detections belonging to a tenant, sorted in descending order (most recent detection listed first). The page number and page size parameters are optional. When the values are not specified, the default values are 1 and 20 respectively.

**Service Endpoint:**

- `/detections/v2?page=m&page_size=n`

Append the following optional query string parameters:

- `page`: The page number to request
- `page_size`: The number of device records to retrieve per page

For example, to return the first page with 100 devices:

`https://protectapi.cylance.com/detections/v2?page=1&page_size=100`

**Method:**

- HTTP/1.1 GET
Request Headers:
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsdetect:list scope encoded

Request:
Append the following optional query string parameters:
- start: Start date-time of the query range
  - Displays events that occurred on or after the specified date and time.
  - Can be used with the end date-time to query a date range.
  - Example: &start=2019-05-04T00:00:00.000Z
- end: End date-time of the query range
  - Displays events that occurred before or on the specified date and time.
  - Can be used with the start date-time to query a date range.
  - Example: &end=2019-05-04T23:00:00.000Z
- severity: Detection severity filter
  - Supports OR filters via multiple queries.
  - Values are Informational, Low, Medium, High.
  - Example: &severity=High
- detection_type: Detection type filter
  - Supports OR filters via multiple queries.
  - This is the DetectionDescription.
  - Example: &detection_type=Powershell Download
- event_number: Event number filter
  - Supports OR filters via multiple queries
  - This is the Phonicld in the API and Detection ID in the Console
  - Filters on partial information
- device: Device name filter
  - Supports OR filters via multiple queries
- status: The status for the detection event.
  - Values are New, In Progress, Follow Up, Reviewed, Done, False Positive
- sort: Sort by the following fields (adding "." in front of the value denotes descending order)
  - Severity
  - OccurrenceTime
  - Status
  - Device
  - PhoneticId
  - Description
  - ReceivedTime

Example to retrieve the first page with up to 100 Detections, with a High severity, and sorted by Occurrence Time:

https://protectapi.cylance.com/detections/v2?page=1&page_size=100&severity=High&sort=OccurrenceTime

Response:

200 OK

```json
{
  "page_number": "1",
  "page_size": "20",
  "total_pages": "1",
  "total_number_of_items": "1",
  "page_items": [
    {
      "PhoneticId": "47E7-0635",
      "Status": "New",
      "Id": "47e70635-b10a-4bde-9ffe-b0258f656a1e",
      "Severity": "Medium",
      "OccurrenceTime": "2019-05-10T22:00:54.289Z",
      "ReceivedTime": "2019-05-10T22:00:56.000Z",
      "Device": {
        "Name": "User-Laptop-A123",
        "CylanceId": "e378dacb-9324-453a-b8c6-5a8406952195"
      },
      "DetectionDescription": "Fileless Powershell Malware"
    }
  ]
}
```

400 Bad Request - Returned for the following reasons:
- The Tenant ID cannot be retrieved from the JWT token
- The page number or page size specified is less than or equal to zero, or the page size is greater than 200

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
404 Not Found - The detection resources page requested doesn't exist.
500 Internal Server Error - An unforeseeable error has occurred.
503 Service Unavailable - Unable to respond at this time, please try again later

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CylanceId</td>
<td>The ID for the device.</td>
</tr>
<tr>
<td>DetectionDescription</td>
<td>The description of the detection.</td>
</tr>
<tr>
<td>Device</td>
<td>The device information contains the device ID and device name.</td>
</tr>
<tr>
<td>Id</td>
<td>The unique ID for the detection.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the device.</td>
</tr>
<tr>
<td>OccurrenceTime</td>
<td>The time when the detection occurred according to the associated endpoint Agent.</td>
</tr>
<tr>
<td>page_number</td>
<td>The page number requested.</td>
</tr>
<tr>
<td>page_size</td>
<td>The page size requested.</td>
</tr>
<tr>
<td>PhoneticId</td>
<td>The easy-to-read version of the ID that is probabilistically unique.</td>
</tr>
<tr>
<td>ReceivedTime</td>
<td>The time when the detection was received by Cylance's cloud services.</td>
</tr>
<tr>
<td>Severity</td>
<td>The criticality of an observance of a detection.</td>
</tr>
<tr>
<td>Status</td>
<td>The status of the detection workflow.</td>
</tr>
<tr>
<td>total_number_of_items</td>
<td>The total number of resources.</td>
</tr>
<tr>
<td>total_pages</td>
<td>The total number of pages that can be retrieved, based on the page size specified.</td>
</tr>
</tbody>
</table>

**Get Detection**

Request a specific detection resource belonging to a tenant. Use Get Detections to obtain the unique detection ID.
Service Endpoint:

- /detections/v2/{detection_id}/details

Example: https://protectapi.cylance.com/detections/v2/f2d6c020-53e2-4300-9005-2e006d9a0f57/details

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsdetect:read scope encoded

Request:

None

Response:

200 OK

```json
{
  "Id": "f2d6c020-53e2-4300-9005-2e006d9a0f57",
  "ActivationTime": "2019-05-10T22:00:54.195Z",
  "AppliedExceptions": [
    {
      "Id": "e669a5f7-2a6b-4860-bb83-47e90283e396",
      "Version": 1
    }
  ],
  "ArtifactsOfInterest": {
    "StateA": {
      {
        "Artifact": {
          "Type": "Process",
          "Uid": "fbDWDraWm1DCa8dmOxX=="
        },
        "Source": "Instigating Process"
      },
      {
        "Artifact": {
          "Type": "Process",
          "Uid": "56ss5cDa88ob0hTQgNOIr=="
        },
        "Source": "Target Process"
      }
    }
  }
}
```
"AssociatedArtifacts": [
  {
    "Uid": "yNhcNbEDtsdCHSFqyOftjv=",
    "Name": "WindowsKernel",
    "Version": "1",
    "ArtifactType": "Sensor"
  },
  {
    "Uid": "fbDWDrQaWm1DCa8dmJkOXx=",
    "ProcessId": 7953,
    "ParentId": 2921,
    "Name": "file.exe",
    "PrimaryImage": {
      "Type": "File",
      "Uid": "yw0mERK08DXy4XCWNdoobB="
    },
    "Owner": {
      "Type": "User",
      "Uid": "bBvAnM5K4Dtp3RSgGq3m1h="
    },
    "CommandLine": "C:\folder\path\file.exe",
    "Description": "Some description of the file",
    "StartDateTime": "2019-05-10T18:11:23.482Z",
    "EndDateTime": "2019-05-10T18:11:23.482Z",
    "PrimaryImageFile": "",
    "PrimaryUser": "",
    "ParentUid": "",
    "Parent": {
      "Type": "Process",
      "Uid": "yNhcNbEDtsdCHSFqyOftjv="
    },
    "ParentStartDateTime": "2019-05-10T18:11:23.482Z",
    "SessionId": -1,
    "IsBeingDebugged": false,
    "ArtifactType": "Process"
  }
],
"Comment": "",
"DetectionRule": {
  "Name": "File Execution",
  "Id": "7c050000-9b99-4f05-a6e0-e598141370d5",
  "PolicyGroup": "TestRuleset",
  "Version": 1,
  "ObjectType": "DetectionRule",
  "Description": "File Execution",
  "Category": "Custom"
},
"Detector": {
  "Name": "OpticsDetector",
  "Version": "2"
},
"Device": {
"CylanceId": "e378dacb-9324-453a-b8c6-5a8406952195",
"Name": "User-Laptop-A123",
"IpAddresses": [ 
"00-00-00-00-00-00",
"123.45.67.89"
],
"LoggedOnUsers": [ 
{ 
"AccountType": "Normal",
"FullName": "", 
"Id": "",
"User": "User-Laptop-A123\Admin"
}
],
"Name": "File Execution",
"OccurrenceTime": "2019-05-03T17:07:09.963Z",
"Product": { 
"Name": "CylanceOPTICS",
"Version": "2.3.2050.930"
},
"PhoneticId": "E669-A5F7",
"ReceivedTime": "2019-05-03T17:07:10Z",
"SchemaVersion": 1,
"Severity": "High",
"SeveritySortLevel": 4,
"Status": "New",
"StatusSortLevel": 1,
"TenantId": "4b1640d2-d563-41cf-94a7-0da1dca6aa98",
"ZoneIds": [],
"Trace": [ 
{ 
"StateName": "fileexecution",
"Event": { 
"Uid": "gwFYUE/5tH4zlTxVsbTHrh==",
"EventCategory": "Process",
"Mode": "UserMode",
"Subcategory": "",
"Source": { 
"Type": "Sensor",
"Uid": "sndMjFxw26PtNedf6ZGeFx=="
},
"OccurrenceTime": "2019-05-03T17:07:09.386Z",
"RegistrationTime": "2019-05-03T17:07:09.386Z",
"EventType": "Start",
"EventDetails": {}
},
"InstigatingProcess": { 
"Type": "Process",
"Uid": "bzfq6/BcV+ilp6GcN6yhYx=="
},
"ProviderUid": "",
"ProviderSequenceId": 11930,
"Targets": [
  {
    "Type": "Process",
    "Uid": "JsDqqMyGW8++RKWP9H8Dx=="
  },
  "GroupUid": ""
],
"Responses": [
  {
    "Status": "New",
    "Comment": "5550-5F91",
    "TenantId": "4b1640d2-d563-41cf-94a7-0da1dca6a998",
    "PhoneticId": "",
    "DetectionId": "98a025bd-7f39-4baa-bfab-4fa1c26f8c0",
    "OccurrenceTime": "2019-05-03T17:07:09.963Z",
    "ActionResults": {
      "additionalProp1": {
        "HandlingResponderVersion": 1,
        "HandlingResponderName": "OpticsResponder",
        "Results": [
          {
            "Status": {
              "Message": "",
              "Code": {
                "Ordinal": 0,
                "Reason": "The action was completed successfully",
                "Name": "Success"
              }
            },
            "ArtifactsOfInterestType": "None"
          }
        ],
        "additionalProp2": {
          "HandlingResponderVersion": ,
          "HandlingResponderName": "",
          "Results": [
            {
              "Status": {
                "Message": "",
                "Code": {
                  "Ordinal": ,
                  "Reason": "",
                  "Name": ""
                }
              },
              "ArtifactsOfInterestType": ""
            }
          ]
        }
      }
    }
  }
]
400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The unique identifier for the detection is not valid

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resource requested does not exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time; try again later.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActivationTime</td>
<td>The time at which this particular detection first started occurring.</td>
</tr>
<tr>
<td>AppliedExceptions</td>
<td>The exceptions that were applied to the detection.</td>
</tr>
<tr>
<td></td>
<td>- Id: The unique identifier for the exception.</td>
</tr>
<tr>
<td></td>
<td>- Version: The version number for the exception.</td>
</tr>
<tr>
<td>ArtifactsOfInterest</td>
<td>The artifact associated with the rule that triggered the exception. This is a dynamic object.</td>
</tr>
<tr>
<td></td>
<td>- Artifact:</td>
</tr>
<tr>
<td></td>
<td>- Type: The type of artifact.</td>
</tr>
<tr>
<td></td>
<td>- Uid: The unique identifier for the artifact.</td>
</tr>
<tr>
<td></td>
<td>- Source: The source for the artifact.</td>
</tr>
<tr>
<td></td>
<td>- StateA: This is the name of the Artifact of Interest.</td>
</tr>
<tr>
<td>AssociatedArtifacts</td>
<td>The list of artifacts that were involved in this detection. These are dynamic objects.</td>
</tr>
<tr>
<td>Comment</td>
<td>The comment on the detection.</td>
</tr>
<tr>
<td>Context</td>
<td>The context of the detection.</td>
</tr>
<tr>
<td>DetectionRule</td>
<td>The description of the rule from which this detection originated.</td>
</tr>
<tr>
<td></td>
<td>- Category: The category of the rule.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| Field Name | Description: The description of the rule.  
Id: The ID of the rule.  
Name: The name of the rule.  
Version: The version of the rule.  
Detector | The description of the plugin that originated the detection.  
Name: The name of the detector.  
Version: The version of the detector.  
Device | A capture of the current state of the device.  
CylanceId: The unique ID for the device.  
Name: The name of the device.  
Id | The unique identifier for the detection.  
InvolvedArtifacts | The artifacts involved in this detection.  
Name | The name of the detection.  
ObjectType | The object type for the detection.  
OccurrenceTime | The time at which the detection occurred.  
PhoneticId | The easy-to-read version of the ID that is probabilistically unique.  
Product | The description of the Cylance product that originated the detection.  
CylanceId: The unique ID for the device.  
Name: The name of the Cylance product.  
Version: The version of the Cylance product.  
ReceivedTime | The time when the detection was received.  
Responses | The responses to the detection.  
Status: The status of the response.  
Comment: The comment on the response.  
TenantId: The tenant ID to which the response belongs.  
PhoneticId: The easy-to-read version of the ID that is probabilistically unique.  
DetectionId: The ID for the detection event that warranted the response.  
OccurrenceTime: The time at which the response actions were taken.  
ActionResults:  
  HandlingResponderVersion: The version of the Responder plugin that performed the response.  
  HandlingResponderName: The name of the Responder plugin that performed the response.  
  Results:  
    Status: The status of the result.
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message</td>
<td>The message of the result.</td>
</tr>
<tr>
<td>Code</td>
<td></td>
</tr>
<tr>
<td>Ordinal</td>
<td>The indicator code for the success of the action.</td>
</tr>
<tr>
<td>Reason</td>
<td>The detailed description explaining the indicator code.</td>
</tr>
<tr>
<td>Name</td>
<td>The friendly name of the status code.</td>
</tr>
<tr>
<td>AssociatedArtifacts</td>
<td>The artifacts upon which the action occurred.</td>
</tr>
<tr>
<td>ResponseRuleId</td>
<td>The ID of the response rule that triggered the response.</td>
</tr>
<tr>
<td>SchemaVersion</td>
<td>The version of the response rule.</td>
</tr>
<tr>
<td>ResponseRuleVersion</td>
<td>The version of the response rule.</td>
</tr>
<tr>
<td>ReceivedTime</td>
<td>The time the response was received.</td>
</tr>
<tr>
<td>ObjectType</td>
<td>The type of the object for the response.</td>
</tr>
</tbody>
</table>

- **SchemaVersion**: The version of the schema to which the object conforms.
- **Severity**: The criticality of an observance of the detection.
- **SeveritySortLevel**: The sort level for the severity.
- **Status**: The status of the detection in the workflow.
- **StatusSortLevel**: The sort level for the status.
- **Trace**: The trace information.
  - **Event**: The CylanceOPTICS Event that triggered the state.
  - **StateName**: The name of a state that was traversed.
- **TenantId**: The ID for the tenant.
- **ZonelIds**: The list of IDs for the zones associated with the detection.

**Get Recent Detections**

Request a count of recent CylanceOPTICS detection resources belonging to a tenant.

**Service Endpoint:**

- `/detections/v2/recent?since={recent_detection_datetime}`

**Example:**

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsdetect:list scope encoded

Request:
None

Response:

200 OK

```json
{
   "num_after": 10,
   "num_unaddressed": 10
}
```

400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The date-time for {recent_detection_datetime} is not valid.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resource requested does not exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time; try again later.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>num_after</td>
<td>The number of detections after the {recent_detection_datetime}.</td>
</tr>
<tr>
<td>num_unaddressed</td>
<td>The number of unaddressed detections after the {recent_detection_datetime}.</td>
</tr>
</tbody>
</table>

Get Detections CSV

Request a list of CylanceOPTICS detection resources belonging to a tenant, in CSV format. Any provided filters will be applied, but limit/offset parameters will not. All detections for the tenant
will be exported.

**Service Endpoint:**

- `/detections/v2/csv`

**Example:** [https://protectapi.cylance.com/detections/v2/csv](https://protectapi.cylance.com/detections/v2/csv)

**Method:**

- HTTP/1.1 GET

**Request Headers:**

- `Accept: application/json`
- `Authorization: Bearer <JWT Token returned by Auth API> with the opticsdetect:list scope encoded`

**Request:**

Append the following optional query string parameters:

- `start`: Start date-time of the query range
- `end`: End date-time of the query range
- `severity`: Detection severity filter
  - Supports OR filters via multiple queries
  - Values are Informational, Low, Medium, High
- `detection_type`: Detection type filter
  - Supports OR filters via multiple queries
- `detected_on`: The detected on filter.
  - Supports OR filters via multiple queries.
- `event_number`: Event number filter
  - Supports OR filters via multiple queries
- `device`: Device name filter
  - Supports OR filters via multiple queries
- `status`: Values are New, In Progress, Follow Up, Reviewed, Done, False Positive
- `page`: The page number to request
  - Default is 1
- **page_size**: The number of detection records to retrieve per page
  - Default is 20
- **sort**: Sort by the following fields (adding "-" in front of the value denotes descending order)
  - Severity
  - OccurrenceTime
  - Status
  - Device
  - PhoneticId
  - Description
  - ReceivedTime

Example to retrieve the first page with up to 100 Detections, with a High severity, and sorted by Occurrence Time:

https://protectapi.cylance.com/detections/v2/csv?page=1&page_size=100&severity=High&sort=OccurrenceTime

**Response:**

200 OK - A proper response displays the requested detection information in CSV format.

400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The one or more of the parameters in the request are not valid.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resource requested does not exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time; try again later.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cylance Id</td>
<td>The ID for the device.</td>
</tr>
<tr>
<td>Device</td>
<td>The name of the device.</td>
</tr>
<tr>
<td>Detected On</td>
<td>The time when the detection occurred according to the associated endpoint Agent.</td>
</tr>
</tbody>
</table>
### Get Detections By Severity

Request a list of CylanceOPTICS aggregated detection resources by severity for a tenant. This is useful for making histograms.

**Service Endpoint:**

- `/detections/v2/severity?start={detection_start_timestamp}&end={detection_end_timestamp}&interval={detection_interval}`

**Example:**


**Method:**

- HTTP/1.1 GET

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsdetect:list scope encoded

**Request:**

Append the following optional query string parameters:

- start: Start date-time of the query range
- end: End date-time of the query range
- interval: The timer interval used for grouping detection resources
  - Format: [0-9] + (year|month|week|day|hour|minute)
  - Example: To group detection resources by one day, use 1d
- detection_type: Detection type filter
  - Supports OR filters via multiple queries
  - This is the DetectionDescription or the Detection Name
- detected_on: The detected on filter.
  - Supports OR filters via multiple queries
- event_number: Event number filter
  - Supports OR filters via multiple queries
  - This is the PhoneticId
- device: Device name filter
  - Supports OR filters via multiple queries
- status: Values are New, In Progress, Follow Up, Reviewed, Done, False Positive

Response:

200 OK

```json
{
  "facet": "Severity",
  "filters": [
    "event_number=F"
  ],
  "results": [
    {
      "detected_on": 
      "counts": {
        "informational": 1,
        "low": 2,
        "medium": 3,
        "high": 4
      }
    }
  ]
}
```

400 Bad Request - Returned for the following reasons:
- The Tenant ID cannot be retrieved from the JWT token
- The one or more of the parameters in the request are not valid.
401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resource requested does not exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time; try again later.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>counts</td>
<td>The number of detections found, grouped by severity - Informational, Medium, and High.</td>
</tr>
<tr>
<td>detected_on</td>
<td>The time when the detection was received by Cylance's cloud services.</td>
</tr>
<tr>
<td>facet</td>
<td>The facet used for the search. This is Severity.</td>
</tr>
<tr>
<td>filters</td>
<td>The list of filters used on the request.</td>
</tr>
</tbody>
</table>

**Update Detection**

Update the status or comment fields for an existing detection for a tenant.

**Service Endpoint:**

- `/detections/v2`

**Example:** https://protectapi.cylance.com/detections/v2

**Method:**

- HTTP/1.1 POST

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsdetect:update scope encoded

**Request with Status:**

```
[  
  {  
    "detection_id": "f2d6c020-53e2-4300-9005-2e006d9a0f57",  
    "field_to_update": { 
      "status": "Done"  
    }  
  }  
]
```
Request with Comment:

```
[
    {
        "detection_id": "f2d6c020-53e2-4300-9005-2e006d9a0f57",
        "field_to_update": {
            "comment": "Add comment"
        }
    }
]
```

**Note:** When creating the request JSON, include the status or comment string, but not both in the same request. Attempting to send the request with the status and comment strings included will result in a 400 bad request error.

**Response:**

200 OK - Successful update

```
[
    {
        "detection_id": "f2d6c020-53e2-4300-9005-2e006d9a0f57",
        "response": "Success"
    }
]
```

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resource to update doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

**Request JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>The comment of the detection.</td>
</tr>
<tr>
<td>status</td>
<td>The status of the detection.</td>
</tr>
<tr>
<td></td>
<td>• Done - All actions are complete for this detection.</td>
</tr>
<tr>
<td></td>
<td>• False Positive - The detection is considered a false positive.</td>
</tr>
<tr>
<td></td>
<td>• Follow Up - This detection requires someone to follow-up on it.</td>
</tr>
<tr>
<td></td>
<td>• In Progress - The detection is currently being reviewed and worked on.</td>
</tr>
<tr>
<td></td>
<td>• New - The detection is new.</td>
</tr>
<tr>
<td></td>
<td>• Reviewed - The detection has been reviewed, but no actions have been taken.</td>
</tr>
</tbody>
</table>
Delete Detection

Soft delete a specific CylanceOPTICS detection resource belonging to a tenant. Use Get Detections to obtain the unique ID for the detection.

Service Endpoint:

- /detections/v2/{detection_id}

Example: https://protectapi.cylance.com/detections/v2/f2d6c020-53e2-4300-9005-2e006d9a0f57

Method:

- HTTP/1.1 DELETE

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsdetect:delete scope encoded

Request:

None

Response:

200 OK

```json
{
  "id": "f2d6c020-53e2-4300-9005-2e006d9a0f57",
  "success": true
}
```

400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The unique identifier for the detection is not valid

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - Returned for the following reasons:

- The JWT Token did not contain the proper scope to perform this action.
- Cannot delete the CylanceOPTICS detection.

404 Not Found - The detection resource requested does not exist.

500 Internal Server Error - An unforeseeable error has occurred.
503 Service Unavailable - Unable to respond at this time; try again later.

**Delete Detections**

Delete CylanceOPTICS detection resources for a specific tenant.

**Service Endpoint:**
- `/detections/v2/`

**Example:** https://protectapi.cylance.com/detections/v2/

**Method:**
- HTTP/1.1 DELETE

**Request Headers:**
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsdetect:delete scope encoded

**Request:**

```
{
   "ids": [
      "f2d6c020-53e2-4300-9005-2e006d9a0f57",
      "23f22a53-e656-4253-8bc5-e40b13e980d4"
   ]
}
```

**Response:**

200 OK

```
{
   "ids": [
      "f2d6c020-53e2-4300-9005-2e006d9a0f57",
      "23f22a53-e656-4253-8bc5-e40b13e980d4"
   ],
   "success": true
}
```

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

500 Internal Server Error - An unforeseeable error has occurred.
Package Deployment API
CylanceOPTICS users can now interact with a hardened Python interpreter that is present locally on each endpoint that is running CylanceOPTICS v2.3.1000 or higher. This new feature allows users to interact with their endpoints in an efficient and technical manner to accomplish tasks on endpoints in an automated fashion. By default, Cylance is supporting 5 capabilities to collect different forensic artifacts from targeted endpoints. These capabilities include:

- Collecting Master File Table (MFT) artifacts from NTFS volumes.
- Collecting entire Windows Registry Hives from endpoints.
- Collecting entire Windows Event Log files from endpoints.
- Collecting Web Browser History Databases from Chrome, Firefox, Internet Explorer, Edge, Opera, and Safari.
- Collecting common Application Execution Records, including Amcache, Prefetch, and Shimcache.

Users can also configure and deploy Custom Packages to conduct custom, scripted actions against endpoints. This allows customers to upload in-house or third-party scripts and applications to Cylance’s cloud services and deploy them to endpoints. This scripting is done via interacting with the local Python interpreter built into CylanceOPTICS, allowing for an easily extensible set of capabilities.

After packages have been deployed and executed on endpoints, users can automatically upload the resulting data to SMB shares or SFTP servers for centralized collection and analysis by other forensic or incident response tools. Users can also configure packages to store the results locally on the endpoints for retrieval at a later time.

The CylanceOPTICS Package Deployment supports up to 20 packages for your organization. Each package has a maximum file size of 15MB.

These capabilities and workflows around the Package Deployment feature are exposed via Cylance’s API.

**Create Package**

Create (add) a new CylanceOPTICS package resource for a tenant.

**Service Endpoint:**

- /packages/v2

**Example:** https://protectapi.cylance.com/packages/v2

**Method:**

- HTTP/1.1 POST
Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticspkgconfig:create scope encoded

Request:

```
{
  "checksum": "bf17366ee3bb8068a9ad70fc9e68496e7e311a055bf4ffeef53cc5d29ccce52",
  "packageDescriptor": {
    "name": "Test Package",
    "description": "This is a test package",
    "examples": [],
    "packageInfo": {
      "fileType": "python",
      "fileName": "hello_world.py",
      "entryPoint": ""
    },
    "version": 1
  }
}
```

Response:

202 Accepted

```
{
  "packageId": "8d1e1033-37dc-4597-b4d8-c56feb6450dc",
  "uploadTo": "https://scalar-api.cylance.com/upload/123456789ABCDEFG",
  "packageUrl": "https://protectapi.cylance.com/123456789ABCDEFG"
}
```

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

500 Internal Server Error - An unforeseeable error has occurred.

Request JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>checksum</td>
<td>The SHA256 hash for the package.</td>
</tr>
<tr>
<td>description</td>
<td>A description of what the package does.</td>
</tr>
</tbody>
</table>
### Field Name | Description
---|---
examples | A list of examples of how to use the package. This information is optional.  
  - description: A description of what the example does.  
  - invocationString: An example of how to invoke the package.
name | The name of the package.
packageInfo | Package level documentation and annotation.  
  - entryPoint: The point of execution for the package.  
  - fileName: The name of the package file.  
  - fileType: The file type of the package.  
  **Note:** Only Python is supported.
version | The version of the package. Must be 1 or higher.

### Response JSON Schema Descriptions

| Field Name | Description |
---|---|
packageId | The unique identifier for the package. |
packageUrl | The URL to retrieve the package (after the actual package has been uploaded). |
uploadTo | The URL used to upload the package. |

### Get Packages

Request a page with a list of packages belonging to a tenant, sorted by the uploaded date, in descending order (most recent uploaded package listed first). The page number and page size parameters are optional. When the values are not specified, the default values are 1 and 20 respectively.

**Service Endpoint:**

- `/packages/v2?page=m&page_size=n`

Append the following optional query string parameters:

- `page`: The page number to request
- `page_size`: The number of device records to retrieve per page

For example, to return the first page with 100 devices:

https://protectapi.cylance.com/packages/v2?page=1&page_size=100
Method:
- HTTP/1.1 GET

Request Headers:
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticspkgconfig:list scope encoded

Request:
Append the following optional query string parameters:
- page: The page number to request.
  - Default is 1.
- page_size: The number of package records to retrieve per page.
  - Default is 20.
- sort: Sort by the following fields (adding "-" in front of a value denotes descending order).
  - packageId: Filter by package ID.
  - uploadedOn: Filter by the uploaded timestamp (in UTC).
  - uploadedBy.id: Filter by the user ID of the user who uploaded the package.
  - uploadedBy.login: Filter by the email of the user who uploaded the package.
  - size: Filter by the size of the package (in bytes).
  - status: Filter by the status of the package upload process.
    - Possible values are: started, success, failed, or timeout.
  - timeout: Filter by the amount of time (in seconds) for the package to upload before the status changes to timeout.
  - packageDescriptor.name: The name of the package.
- category: Filter by the package category.
  - Possible values are: custom or cylance.

Example to retrieve the first page with up to 100 Packages, sorted by a success status:
https://protectapi.cylance.com/packages/v2?page=1&page_size=100&status=success

Response:
200 OK
400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token
- The page number or page size specified is less than or equal to zero, or the page size is greater than 200

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The package resources page requested doesn't exist.
500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>category</td>
<td>The category of the package.</td>
</tr>
<tr>
<td>downloadUrl</td>
<td>The URL to download the package from.</td>
</tr>
<tr>
<td>packageDescriptor</td>
<td>The package metadata, provided by the user.</td>
</tr>
<tr>
<td>playbookCount</td>
<td>The number of playbooks to which the package is associated.</td>
</tr>
<tr>
<td>size</td>
<td>The size of the package (in bytes).</td>
</tr>
<tr>
<td>status</td>
<td>The status of the package in the upload process.</td>
</tr>
<tr>
<td>timeout</td>
<td>The amount of time (in seconds) for the package to upload before the status</td>
</tr>
<tr>
<td></td>
<td>changes to timeout.</td>
</tr>
<tr>
<td>uploadedBy</td>
<td>The unique identifier of the user who uploaded the package.</td>
</tr>
</tbody>
</table>

- category: Values are custom or cylance.

- packageDescriptor:
  - description: The description of the package.
  - examples: The list of examples of how to use the package.
    - invocationString: An example of how to invoke the package.
    - description: A description of what the example does.
  - name: The name of the package.
  - packageId: The unique ID for the package.
  - packageInfo: The package level documentation / annotation.
    - fileType: The file type of the package. Only Python is supported.
    - fileName: The name of the package file.
    - entryPoint: The point of execution for the package.
  - version: The version of the package.

- uploadedBy:
  - id: The unique ID for the user.
  - login: The email address of the user.
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uploadedOn</td>
<td>The date and time (in UTC) when the package was uploaded.</td>
</tr>
</tbody>
</table>

## Get Package

Request a specific package resource belonging to a tenant. Use Get Packages to obtain the unique package ID.

### Service Endpoint:

- `/packages/v2/{unique_package_id}`

### Example:

https://protectapi.cylance.com/packages/v2/e378dacb-9324-453a-b8c6-5a8406952195

### Method:

- HTTP/1.1 GET

### Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticspkgconfig:read scope encoded

### Request:

None

### Response:

200 OK

```json
{
  "packageId": "e378dacb-9324-453a-b8c6-5a8406952195",
  "uploadedOn": "2019-03-19T05:25:45Z",
  "uploadedBy": {
    "id": "a2c0ac7a-a63d-4583-b646-ae10db9c9769",
    "login": "testuser@email.com"
  },
  "size": 1000,
  "status": "success",
  "timeout": 300,
  "downloadUrl": "https://content.cylance.com/123456789ABCDEFG",
  "category": "custom",
  "playbookCount": 0,
  "packageDescriptor": {
    "name": "Cylance - Browser History",
    "description": "Browser History"
  }
}```
"examples": [  
  {  
    "invocationString": "-browser Name",  
    "description": "Browser Name"  
  }  
],  
"packageInfo": {  
  "fileType": "csharp",  
  "fileName": "BrowserHistory",  
  "entryPoint": "Cylance.Foundations.Refract.RefractM.BrowserHistory"  
},  
"version": 1 
)

400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token.
- The unique identifier for the package is not valid.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The package resource requested does not exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, try again later.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>category</td>
<td>The category of the package. Values are custom or cylance.</td>
</tr>
<tr>
<td>downloadUrl</td>
<td>The URL to download the package from.</td>
</tr>
<tr>
<td>packageDescriptor</td>
<td>The package metadata, provided by the user.</td>
</tr>
<tr>
<td></td>
<td>- description: The description of the package.</td>
</tr>
<tr>
<td></td>
<td>- examples: A list of examples of how to use the package.</td>
</tr>
<tr>
<td></td>
<td>- description: A description of what the example does.</td>
</tr>
<tr>
<td></td>
<td>- invocationString: An example of how to invoke the package.</td>
</tr>
<tr>
<td></td>
<td>- name: The name of the package.</td>
</tr>
<tr>
<td></td>
<td>- packageInfo: Package level documentation / annotation.</td>
</tr>
<tr>
<td></td>
<td>- entryPoint: The point of execution for the package.</td>
</tr>
<tr>
<td></td>
<td>- fileName: The name of the package file.</td>
</tr>
<tr>
<td></td>
<td>- fileType: The file type of the package. Only Python is supported.</td>
</tr>
<tr>
<td></td>
<td>- packageId: The unique identifier for the package.</td>
</tr>
</tbody>
</table>
Field Name | Description
--- | ---
version | The version of the package.
packageId | The unique identifier for the package.
playbookCount | The number of playbooks to which the package is associated.
size | The size of the package, in bytes.
status | The status of the package in the upload process. Statuses are started, success, failed, and timeout.
timeout | The amount of time (in seconds) for a package upload before the status changes to timeout.
uploadedBy | The unique identifiers of the user who uploaded the package.
  - id: The unique ID for the user.
  - login: The email address of the user.
uploadedOn | The date and time (in UTC) when the package was uploaded.

## Delete Package

Delete a specific package resource belonging to a tenant. Use Get Packages to obtain the unique package ID.

**Service Endpoint:**

- /packages/v2/{unique_package_id}

**Example:** https://protectapi.cylance.com/packages/v2/e378dacb-9324-453a-b8c6-5a8406952195

**Method:**

- HTTP/1.1 DELETE

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticspkgconfig:delete scope encoded

**Request:**

None

**Response:**

204 No Content - Package deleted
400 Bad Request - Returned for the following reasons:

- The Tenant ID cannot be retrieved from the JWT token.
- The unique identifier for the package is not valid.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - Returned for the following reasons:

- The JWT Token did not contain the proper scope to perform this action.
- Cannot delete the CylanceOPTICS package.

404 Not Found - The package resource requested does not exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, try again later.

Create Package Execution

Create (add) a new CylanceOPTICS package execution resource for a specific tenant, which triggers a package to execute on the device or on devices in a specific zone.

Service Endpoint:

- /packages/v2/executions

Example: https://protectapi.cylance.com/packages/v2/executions

Method:

- HTTP/1.1 POST

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticspkgdeploy:create scope encoded

Notes:

- To create a package execution, either provide a list of devices or a list of zones.
- For "devices" or "zones", use the SHA256 format. Example: "E378DACB9324453AB8C65A8406952195"
- For "package" use the UUID4 format. Example: "61d8944e-d900-4724-87eb-d10078b90a41"

Request with Device ID and Destination Local:
Request with Zone ID and Destination FTP:

```
{
  "execution": {
    "name": "Package Execution",
    "target": {
      "zones": [
        "D27FF5C45C0D4F56A00DA1FB297E440E"
      ],
      "destination": "ftp.someaddress.com",
      "packageExecutions": [
        {
          "arguments": [
            "-browser ALL"
          ],
          "package": "61d8944e-d900-4724-87eb-d10078b90a41"
        }
      ],
      "keepResultsLocally": false
    }
  }
}
```

Response with Device ID and Destination Local:

```
202 Accepted

{
  "name": "Package Execution",
  "target": {
    "devices": [
      "E378DACB9324453AB8C65A8406952195"
    ],
    "destination": 
"Package Execution",
"target": {
  "devices": [ "E378DACB9324453AB8C65A8406952195"
спроведення виїмки виконано
"device": 
"Package Execution",
"target": {
  "devices": [ "E378DACB9324453AB8C65A8406952195"
спроведення виїмки виконано
"destination": "}
"packageExecutions": [
  {
    "arguments": [
      "-browser ALL"
    ],
    "package": "61d8944e-d900-4724-87eb-d10078b90a41"
  }
],
"keepResultsLocally": true
}
```

Cylance User API Guide, v2.0 rev24, February 2020 | 197
Response with Zone ID and Destination FTP:

202 Accepted

"name": "Package Execution",
"target": {
  "zones": [
    "d27ff5c4-5c0d-4f56-a00d-a1fb297e440e"
  ]
},
"destination": "ftp.someaddress.com",
"packageExecutions": [ 
  { 
    "arguments": [ 
      "-browser ALL"
    ],
    "package": "61d8944e-d900-4724-87eb-d10078b90a41"
  }
],
"id": "",
"createdAt": "2019-04-25T18:02:28.244Z",
"createdBy": { 
  "id": "3c156323-d910-4947-bff1-9ed77e87f529",
  "login": "API Access"
},
"deviceStatuses": { 
  "acked": 0,
  "successed": 0,
  "failed": 0
},
"deviceCount": 1
}
"deviceStatuses": {
  "acked": 0,
  "successed": 0,
  "failed": 0
},
"deviceCount": 1

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

500 Internal Server Error - An unforeseeable error has occurred.

Request JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>destination</td>
<td>The FTP, SFTP, or SAMBA URL for saving the results. If saving the results to the local disk drive, leave this empty (&quot;&quot;&quot;).</td>
</tr>
<tr>
<td>keepResultsLocally</td>
<td>The setting to save the results to the local disk drive. If true, the results are saved to file:///[Cylance Data Directory]/Optics.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the execution.</td>
</tr>
<tr>
<td>packageExecutions</td>
<td>The list of packages to execute.</td>
</tr>
<tr>
<td></td>
<td>- arguments: The list of arguments for the package. See examples from packageDescriptor.</td>
</tr>
<tr>
<td></td>
<td>- package: The packageID for the package.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> As an example, you can get the packageID for a package through the “Get Packages” query.</td>
</tr>
<tr>
<td>target</td>
<td>The devices or zones to execute the packages against.</td>
</tr>
<tr>
<td></td>
<td>- devices: The list of device IDs to execute the packages against.</td>
</tr>
<tr>
<td></td>
<td>- zones: The list of zone IDs to execute the packages against.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Target devices only or zones only per request, not both.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Device IDs and zone IDs must be uppercase letters and no hyphens.</td>
</tr>
</tbody>
</table>

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>createdAt</td>
<td>The date and time (in UTC) when the execution was requested.</td>
</tr>
<tr>
<td>createdBy</td>
<td>The user who requested the execution.</td>
</tr>
<tr>
<td></td>
<td>- id: The unique ID for the user.</td>
</tr>
<tr>
<td></td>
<td>- login: The email address of the user.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>destination</td>
<td>The FTP, SFTP, or SAMBA URL for saving the results.</td>
</tr>
<tr>
<td>deviceCount</td>
<td>The number of online devices at the moment the package execution request was made.</td>
</tr>
<tr>
<td>deviceStatuses</td>
<td>The statuses of the package executions on the devices.</td>
</tr>
<tr>
<td></td>
<td>- acked (acknowledged): The number of devices that receive the package execution command but have not yet responded.</td>
</tr>
<tr>
<td></td>
<td>- failed: The number of devices that failed to execute the packages.</td>
</tr>
<tr>
<td></td>
<td>- succeeded: The number of devices that have successfully executed the packages.</td>
</tr>
<tr>
<td>id</td>
<td>The ID of the execution resource.</td>
</tr>
<tr>
<td>keepResultsLocally</td>
<td>The setting to save the results to the local disk drive. If true, the results are saved to file?://{Cylance Data Directory}/Optics.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the execution.</td>
</tr>
<tr>
<td>packageExecutions</td>
<td>The list of packages to execute.</td>
</tr>
<tr>
<td></td>
<td>- arguments: The list of arguments for the package. See examples from packageDescriptor.</td>
</tr>
<tr>
<td></td>
<td>- package: The packageID for the package.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> As an example, you can get the packageID for a package through the “Get Packages” query.</td>
</tr>
<tr>
<td>target</td>
<td>The devices and/or zones to execute the packages against.</td>
</tr>
<tr>
<td></td>
<td>- devices: The list of device IDs to execute the packages against.</td>
</tr>
<tr>
<td></td>
<td>- zones: The list of zone IDs to execute the packages against.</td>
</tr>
</tbody>
</table>

**Get Package Executions**

Request a page with a list of package executions belonging to a tenant, sorted by the uploaded date, in descending order (most recent uploaded package execution listed first). The page number and page size parameters are optional. When the values are not specified, the default values are 1 and 20 respectively.

**Service Endpoint:**

- `/packages/v2/executions?page=m&page_size=n`

Example:

https://protectapi.cylance.com/packages/v2/executions?page=1&page_size=100

**Method:**
HTTP/1.1 GET

Request Headers:
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticspkgdeploy:list scope encoded

Request:
Append the following optional query string parameters:
- page: The page number to request.
  - Default is 1.
- page_size: The number of package records to retrieve per page.
  - Default is 20.
- sort: Sort by the following fields (adding "-" in front of a value denotes descending order).
  - id
  - name
  - createdAt
  - createdBy.Id
  - createdBy.Login
  - deviceCount
- id: Filter by unique ID of the execution.
- name: Filter by name of the execution.
- createdAt: Filter by date and time (in UTC) when the execution was requested.
- createdBy.id: Filter by ID of the user who requested the execution.
- createdBy.login: Filter by the email address of the user who requested the execution.
- deviceCount: Filter by the number of online devices at the moment the package execution request was made.

Example to retrieve the first page with up to 100 Detections, filtered by ID, and sorted by ID:
https://protectapi.cylance.com/packages/v2/executions?page=1&page_size=100&id=1C04-1C2D&sort=id

Response:
200 OK
400 Bad Request - Returned for the following reasons:

- The tenant ID could not be retrieved from the JWT token specified in the Authorization header.
- The page number or page size specified are less than or equal to zero.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The package resources page requested does not exist.
500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, try again later.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>createdAt</td>
<td>The date and time (in UTC) when the execution was requested.</td>
</tr>
<tr>
<td>createdBy</td>
<td>The user who requested the execution.</td>
</tr>
<tr>
<td></td>
<td>- id: The unique identifier of the user who requested the execution.</td>
</tr>
<tr>
<td></td>
<td>- login: The email address of the user who requested the execution.</td>
</tr>
<tr>
<td>destination</td>
<td>The FTP, SFTP, or SAMBA URL for saving the results.</td>
</tr>
<tr>
<td>deviceCount</td>
<td>The number of online devices at the moment the package execution request was made.</td>
</tr>
<tr>
<td>deviceStatuses</td>
<td>The statuses of the package executions on the device.</td>
</tr>
<tr>
<td></td>
<td>- acked (acknowledged): The number of devices that received the package.</td>
</tr>
<tr>
<td></td>
<td>- failed: The number of devices that failed to execute the packages.</td>
</tr>
<tr>
<td></td>
<td>- succeeded: The number of devices that have successfully executed the packages.</td>
</tr>
<tr>
<td>id</td>
<td>The unique identifier of the execution resource.</td>
</tr>
<tr>
<td>keepResultsLocally</td>
<td>The setting to save the results to the local disk drive. If true, the results are saved to file://[/Cylance Data Directory]/Optics.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the execution.</td>
</tr>
<tr>
<td>packageExecutions</td>
<td>The list of packages to execute.</td>
</tr>
<tr>
<td></td>
<td>- arguments: The list of arguments for the package. See examples from packageDescriptor.</td>
</tr>
<tr>
<td></td>
<td>- package: The URL from which to download the package resource.</td>
</tr>
<tr>
<td>page_number</td>
<td>The page number requested.</td>
</tr>
<tr>
<td>page_size</td>
<td>The page size requested.</td>
</tr>
<tr>
<td>target</td>
<td>The devices and/or zones to execute the packages against.</td>
</tr>
<tr>
<td></td>
<td>- devices: The list of device IDs to execute the packages against.</td>
</tr>
<tr>
<td></td>
<td>- The list of zone IDs to execute the packages against.</td>
</tr>
<tr>
<td>total_number_of_items</td>
<td>The total number of resources.</td>
</tr>
<tr>
<td>total_pages</td>
<td>The total number of pages that can be retrieved based on the page size specified.</td>
</tr>
</tbody>
</table>
Get Package Execution

Request a specific package execution resource belonging to a tenant. Use Get Package Executions to obtain the unique package execution ID.

Service Endpoint:

- /packages/v2/executions/{unique_execution_id}

Example:
https://protectapi.cylance.com/packages/v2/executions/abebf88c-f283-4edc-834b-aa8ad3bc682c

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticspkgdepl oy:read scope encoded

Request:

None

Response:

200 OK

```json
{
  "name": "Package Execution",
  "target": {
    "devices": [
      "e378dacb-9324-453a-b8c6-5a8406952195"
    ],
    "destination": "",
    "packageExecutions": [
      {
        "arguments": [
          "-browser ALL"
        ],
        "packages": "https://scalar-api.cylance.com/packages"
      }
    ],
    "keepResultsLocally": true,
    "id": "abebf88c-f283-4edc-834b-aa8ad3bc682c",
    "createdAt": "2019-04-25T18:02:28.244Z",
    "createdBy": {
      "name": "Cylance User API Guide, v2.0 rev24, February 2020 | 204"}
"id": "a2c0ac7a-a63d-4583-b646-ae10db9c9769",
"login": "API Access"
},
"deviceStatuses": {
  "acked": 1,
  "succeeded": 0,
  "failed": 0
},
"deviceCount": 1
}

400 Bad Request - Returned for the following reasons:

- The tenant ID could not be retrieved from the JWT token specified in the Authorization header.
- The unique identifier for the execution is not valid.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
404 Not Found - The execution resources page requested does not exist.
500 Internal Server Error - An unforeseeable error has occurred.
503 Service Unavailable - Unable to respond at this time, try again later.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>createdAt</td>
<td>The date and time (in UTC) when the execution was requested.</td>
</tr>
<tr>
<td>createdBy</td>
<td>The user who requested the execution.</td>
</tr>
<tr>
<td></td>
<td>id: The unique ID for the user.</td>
</tr>
<tr>
<td></td>
<td>login: The email address of the user.</td>
</tr>
<tr>
<td>destination</td>
<td>The FTP, SFTP, or SAMBA URL for saving the results.</td>
</tr>
<tr>
<td>deviceCount</td>
<td>The number of online devices at the moment the package execution request was made.</td>
</tr>
<tr>
<td>deviceStatuses</td>
<td>The statuses of the package executions on the devices.</td>
</tr>
<tr>
<td></td>
<td>acked (acknowledged): The number of devices that received the package execution.</td>
</tr>
<tr>
<td></td>
<td>failed: The number of devices that failed to execute the packages.</td>
</tr>
<tr>
<td></td>
<td>succeeded: The number of devices that have successfully executed the packages.</td>
</tr>
<tr>
<td>id</td>
<td>The ID of the execution resource.</td>
</tr>
<tr>
<td>keepResultsLocally</td>
<td>The setting to save the results to the local disk drive. If true, the results are saved</td>
</tr>
</tbody>
</table>
### Field Name | Description
--- | ---
to file://(Cylance Data Directory)/Optics. | name | The name of the execution.
packageExecutions | The list of packages to execute.  
  - arguments: The list of arguments for the package. See examples from packageDescriptor.  
  - package: The URL to download the package resource from.
target | The devices and/or zones to execute the package against.  
  - devices: The list of device IDs to execute the packages against.  
  - zones: The list of zone IDs to execute the packages against.

## Delete Package Execution

Delete a specific package resource belonging to a tenant. Use Get Package Executions to obtain the unique package ID.

**Service Endpoint:**

- `/packages/v2/executions/{unique_execution_id}`

**Example:**

https://protectapi.cylance.com/packages/v2/executions/abebf88c-f283-4edc-834b-aa8ad3bc682c

**Method:**

- HTTP/1.1 DELETE

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticspkgdeploy:delete scope encoded

**Request:**

None

**Response:**

200 OK

```json
{
  "name": "Package Execution",
  "target": {
```
400 Bad Request - Returned for the following reasons:

- The tenant ID could not be retrieved from the JWT token specified in the Authorization header.
- The unique identifier for the execution is not valid.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - Returned for the following reasons:

- The JWT Token did not contain the proper scope to perform this action.
- Cannot delete the CylanceOPTICS package.

404 Not Found - The execution resources page requested does not exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, try again later.
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>createdAt</td>
<td>The date and time (in UTC) when the execution was requested.</td>
</tr>
<tr>
<td>createdBy</td>
<td>The user who requested the execution.</td>
</tr>
<tr>
<td>id</td>
<td>The unique ID for the user.</td>
</tr>
<tr>
<td>login</td>
<td>The email address of the user.</td>
</tr>
<tr>
<td>destination</td>
<td>The FTP, SFTP, or SAMBA URL for saving the results.</td>
</tr>
<tr>
<td>deviceCount</td>
<td>The number of online devices at the moment the package execution request was made.</td>
</tr>
<tr>
<td>deviceStatuses</td>
<td>The statuses of the package executions on the devices.</td>
</tr>
<tr>
<td>id</td>
<td>The ID of the execution resource.</td>
</tr>
<tr>
<td>keepResultsLocally</td>
<td>The setting to save the results to the local disk drive. If true, the results are saved to file://(Cylance Data Directory)/Optics.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the execution.</td>
</tr>
<tr>
<td>packageExecutions</td>
<td>The list of packages to execute.</td>
</tr>
<tr>
<td>target</td>
<td>The devices and/or zones to execute the package against.</td>
</tr>
<tr>
<td>id</td>
<td>The unique ID for the user.</td>
</tr>
<tr>
<td>login</td>
<td>The email address of the user.</td>
</tr>
<tr>
<td>destination</td>
<td>The FTP, SFTP, or SAMBA URL for saving the results.</td>
</tr>
<tr>
<td>deviceCount</td>
<td>The number of online devices at the moment the package execution request was made.</td>
</tr>
<tr>
<td>deviceStatuses</td>
<td>The statuses of the package executions on the devices.</td>
</tr>
<tr>
<td>id</td>
<td>The ID of the execution resource.</td>
</tr>
<tr>
<td>keepResultsLocally</td>
<td>The setting to save the results to the local disk drive. If true, the results are saved to file://(Cylance Data Directory)/Optics.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the execution.</td>
</tr>
<tr>
<td>packageExecutions</td>
<td>The list of packages to execute.</td>
</tr>
<tr>
<td>target</td>
<td>The devices and/or zones to execute the package against.</td>
</tr>
</tbody>
</table>

- **deviceStatuses**
  - **acked (acknowledged):** The number of devices that received the package execution.
  - **failed:** The number of devices that failed to execute the packages.
  - **succeeded:** The number of devices that have successfully executed the packages.

- **packageExecutions**
  - **arguments:** The list of arguments for the package. See examples from packageDescriptor.
  - **package:** The URL to download the package resource from.
Detection Rule API
The CylanceOPTICS Detection Rules API allows users to create or update rules to help monitor an organization for security threats or anomalous behavior. The flexibility of Detection Rules allows users to monitor for broad behavior characteristics (for example, files being created with certain naming patterns) or search for a targeted series of events (for example, a process with a certain file signature thumbprint that then creates files and initiates network connections).

The CylanceOPTICS Detection Rules API includes:

- Getting the content of a Detection Rule.
- Getting a list of Detection Rules for a tenant.
- Getting a list of Detection Rules as a CSV file.
- Validating a Detection Rule.
- Creating a Detection Rule.
- Updating a Detection Rule.
- Deactivating (or soft deleting) a Detection Rule.
- Getting a natural language representation of a Detection Rule.
- Getting a count of how many Detection Rules exist in a tenant.

### Get Detection Rule List

Retrieve a list of Detection rules available in a tenant.

**Service Endpoint:**

```
/rules/v2?page=m&page_size=n
```

Append the following optional query string parameters:

- **page:** The page number to request
- **page_size:** The number of device records to retrieve per page

For example, to return the first page with 100 devices:

```
https://protectapi.cylance.com/rules/v2?page=1&page_size=100
```

**Method:**

- HTTP/1.1 GET

**Request Headers:**

- **Accept:** application/json
- **Authorization:** Bearer &lt;JWT Token returned by Auth API&gt; with the opticsrule:list scope encoded
Request:
None

Response:
200 OK

```json
{
  "page_size": "1",
  "total_pages": "1",
  "page_items": [
    {
      "Id": "008ece50-49af-472a-b0d8-3c3700883738",
      "Category": "Cylance MITRE ATT&CK Rules",
      "Name": "Gatekeeper Bypass (MITRE)",
      "Version": 1,
      "RulesetCount": 1,
      "LastModified": "2018-11-20T17:58:49Z",
      "Severity": "Medium",
      "DeviceCount": 1,
      "OperatingSystems": [
        {
          "Name": "macOS",
        }
      ],
      "ModifiedBy": {
        "login": "Threat Research",
        "id": "a2c0ac7a-a63d-4583-b646-ae10db9c9769"
      },
      "Description": "Detects bypass of Gatekeeper by non-root user."
    }
  ],
  "total_number_of_items": 1,
  "page_number": 1
}
```

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resources page requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later

Response JSON Schema Descriptions
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>The category of rule grouping that the Detection Rule belongs to. Possible values include:</td>
</tr>
<tr>
<td></td>
<td>- Custom: Custom rules that users have uploaded to a tenant.</td>
</tr>
<tr>
<td></td>
<td>- Cylance Rules: Cylance-official rules.</td>
</tr>
<tr>
<td></td>
<td>- Cylance Experimental: Cylance rules that are deemed to be experimental in their nature.</td>
</tr>
<tr>
<td>Description</td>
<td>The description of the Detection Rule.</td>
</tr>
<tr>
<td>DeviceCount</td>
<td>The number of devices that have the Detection Rule applied.</td>
</tr>
<tr>
<td>Id</td>
<td>The unique ID of the Detection Rule.</td>
</tr>
<tr>
<td>LastModified</td>
<td>The timestamp (in UTC) of the last time that the Detection Rule was modified.</td>
</tr>
<tr>
<td>ModifiedBy</td>
<td>An object detailing the last user to modify the Detection Rule. It includes the following fields:</td>
</tr>
<tr>
<td></td>
<td>- id: The unique ID of the user who modified the Detection Rule.</td>
</tr>
<tr>
<td></td>
<td>- login: The email address of the user who modified the Detection Rule.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the Detection Rule.</td>
</tr>
<tr>
<td>OperatingSystems</td>
<td>An object detailing the operating systems that the Detection Rule can be applied to. It will include the &quot;name&quot; field. This can consist of:</td>
</tr>
<tr>
<td></td>
<td>- &quot;Windows&quot;</td>
</tr>
<tr>
<td></td>
<td>- &quot;MacOS&quot;</td>
</tr>
<tr>
<td>page_number</td>
<td>The current page number of results.</td>
</tr>
<tr>
<td>page_size</td>
<td>The number of items on the page.</td>
</tr>
<tr>
<td>RulesetCount</td>
<td>The number of Detection Rule Sets that have the Detection Rule enabled.</td>
</tr>
<tr>
<td>Severity</td>
<td>The severity assigned to the Detection Rule. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>- High</td>
</tr>
<tr>
<td></td>
<td>- Medium</td>
</tr>
<tr>
<td></td>
<td>- Low</td>
</tr>
<tr>
<td></td>
<td>- Informational</td>
</tr>
<tr>
<td>total_number_of_</td>
<td>The total number of Detection Rules in the tenant.</td>
</tr>
<tr>
<td>items</td>
<td></td>
</tr>
<tr>
<td>total_pages</td>
<td>The total number of pages of this size.</td>
</tr>
<tr>
<td>Version</td>
<td>The version of the Detection Rule.</td>
</tr>
</tbody>
</table>
Get Detection Rule CSV List

Retrieve a CSV file where every line represents a Detection Rule available in the tenant.

**Service Endpoint:**

- /rules/v2/csv

**Example:** https://protectapi.cylance.com/rules/v2/csv

**Method:**

- HTTP/1.1 GET

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsrule:list scope encoded

**Request:**

None

**Response:**

200 OK - A proper response displays the requested detection information in CSV format.

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such resource found.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>The category to which the Detection Rule belongs.</td>
</tr>
<tr>
<td>Description</td>
<td>The description of the Detection Rule.</td>
</tr>
<tr>
<td>Device Count</td>
<td>The number of devices that have the Detection Rule applied.</td>
</tr>
<tr>
<td>Id</td>
<td>The unique ID of the Detection Rule.</td>
</tr>
<tr>
<td>Last Modified</td>
<td>The timestamp (in UTC) of the last time that the Detection Rule was modified.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Modified By</td>
<td>The email address of the user who last modified the Detection Rule.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the Detection Rule.</td>
</tr>
<tr>
<td>Ruleset Count</td>
<td>The number of Detection Rule Sets that have the Detection Rule enabled.</td>
</tr>
<tr>
<td>Severity</td>
<td>The severity of the Detection Rule.</td>
</tr>
<tr>
<td>Version</td>
<td>The version of the Detection Rule.</td>
</tr>
</tbody>
</table>

**Get Detection Rule**

Retrieve the content of a Detection Rule in its native JSON structure.

**Service Endpoint:**

- `/rules/v2/{rule_id}`

For example: https://protectapi.cylance.com/rules/v2/008ece50-49af-472a-b0d8-3c3700883738

**Method:**

- HTTP/1.1 GET

**Request Headers:**

- Accept: application/json
- Authorization: Bearer &lt;JWT Token returned by Auth API&gt; with the opticsdetect:read scope encoded

**Request:**

None

**Response:**

200 OK

```json
{
    "MaximumConcurrentActivations": 1,
    "ActivationLifetimeLimit": "00:10:00",
    "TerminateActiveDfaIfActivatingProcessesEnd": false,
    "ActivationCanUtilizeDeviceStateEvents": false,
    "AllowMultipleActivationsPerContext": true,
    "OperatingSystems": [
        {
            "Name": "macOS"
        }
    ]
}
```
"States": [  
  {  
    "Name": "NonrootAppleScript",
    "Scope": "Global",
    "Function": "(!a&(b))",
    "FieldOperators": {  
      "a": {  
        "Type": "Contains",
        "Operands": [  
          {  
            "Source": "InstigatingProcessOwner",
            "Data": "Name"
          },  
          {  
            "Source": "Literal",
            "Data": "root"
          ]
        }  
      },  
      "OperandType": "String",
      "Options": {  
        "IgnoreCase": true
      }
    },  
    "b": {  
      "Type": "ContainsAny",
      "Operands": [  
        {  
          "Source": "TargetProcess",
          "Data": "CommandLine"
        },  
        {  
          "Source": "LiteralSet",
          "Data": [  
            "osascript",
            "/usr/bin/osascript"
          ]
        ]
      },  
      "OperandType": "String",
      "Options": {  
        "IgnoreCase": true
      }
    }
  },  
  "ActivationTimeLimit": "-0:00:00.001",
  "Actions": [  
    {  
      "Type": "AOI",
      "ItemName": "InstigatingProcess",
      "Position": "PostActivation"
    }
  ]
]


```json
{
    "Type": "AOI",
    "ItemName": "InstigatingProcessOwner",
    "Postion": "PostActivation"
},
{
    "Type": "AOI",
    "ItemName": "TargetProcess",
    "Position": "PostActivation"
}
,"HarvestContributingEvent": true,
"Filters": [
{
    "Type": "Event",
    "Data": {
        "Category": "Process",
        "SubCategory": "",
        "Type": "Start"
    }
}
],
"Paths": [
{
    "StateNames": [
        "NonrootAppleScript"
    ]
}
],
"ObjectType": "DetectionRule",
"Name": "AppleScript (MITRE)",
"Id": "008ece50-49af-472a-b0d8-3c3700883738",
"Version": 1,
"SchemaVersion": 1,
"Description": "A non-root user runs an AppleScript file from the command line via osascript (MITRE 1155)",
"Tags": [
    "Mitre ATT&CK",
    "Execution",
    "Lateral Movement"
],
"RuleSource": "Cylance",
"RuleSourceGrouping": "Mitre Experimental",
"Severity": "Medium",
"Product": {
    "Name": "CylanceOPTICS"
}
"Plugin": {
    "Name": "OpticsDetector"
}
}
```
"NotValidBefore": "1970-01-01T00:00:00.000Z",
"NotValidAfter": "9999-12-31T23:59:59.999Z"
}

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resources page requested doesn’t exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActivationCanUtilizeDeviceStateEvents</td>
<td>Indicates if state events (historical rundowns) should be considered when evaluating for matches.</td>
</tr>
<tr>
<td>ActivationLifetimeLimit</td>
<td>The amount of time a rule is active. If the rule has been active past this duration, then the instance of the rule will be removed.</td>
</tr>
<tr>
<td>AllowMultipleActivationsPerContext</td>
<td>Indicates if the rule can be activated multiple times, simultaneously.</td>
</tr>
<tr>
<td>Description</td>
<td>The description for the Detection Rule.</td>
</tr>
<tr>
<td>Id</td>
<td>The unique identifier for the Detection Rule.</td>
</tr>
<tr>
<td>MaximumConcurrentActivations</td>
<td>Indicates the maximum number of concurrently executing instances of this rule.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the Detection Rule.</td>
</tr>
<tr>
<td>NotValidAfter</td>
<td>The date and time (in UTC) after which the Detection Rule is not valid.</td>
</tr>
<tr>
<td>NotValidBefore</td>
<td>The date and time (in UTC) before which the Detection Rule is not valid.</td>
</tr>
<tr>
<td>ObjectType</td>
<td>The type of object defined in this rule.</td>
</tr>
<tr>
<td></td>
<td>- DetectionRule</td>
</tr>
<tr>
<td></td>
<td>- ResponseRule</td>
</tr>
<tr>
<td>OperatingSystems</td>
<td>The affected operating systems.</td>
</tr>
<tr>
<td></td>
<td>- Name: The name of the type of operating system (like Windows, macOS, or Linux).</td>
</tr>
<tr>
<td>Paths</td>
<td>Defines the paths by which this Deterministic Finite</td>
</tr>
</tbody>
</table>
### Field Name | Description
--- | ---
Automata (DFA) | can be iterated.
Plugin | The CylanceOPTICS plugin associated with the Detection Rule.
Product | The name of the product associated with the Detection Rule.
RuleSource | The source of the rule. For example: Cylance.
RuleSourceGrouping | The classification or designator for the rule source. For example: Optics.
SchemaVersion | The version of the schema.
Severity | The severity assigned to the Detection Rule. Possible values are:  
- High  
- Medium  
- Low  
- Informational
States | The list of all available states. If no paths are specified, the states are transitioned in the order they are specified.  
See the [Custom Rule Guide] on page 331 for information about States.
Tags | A list of tags associated with the Detection Rule.
TerminateActiveDfaIfActivatingProcessesEnd | If the activating process (and, if applicable, all other processes that have been absorbed as activating processes end) then this
Version | The version of the Detection Rule.

---

**Validate Detection Rule**

Allows a user to validate a Detection Rule's JSON by sending the native JSON structure of a Detection Rule to a validation service.

**Service Endpoint:**

- /rules/v2/validate

**Example:** [https://protectapi.cylance.com/rules/v2/validate](https://protectapi.cylance.com/rules/v2/validate)

**Method:**
- **HTTP/1.1 POST**

**Request Headers:**

- **Accept**: application/json
- **Authorization**: Bearer <JWT Token returned by Auth API> with the opticsrule:read scope encoded

**Request:**

```json
{
    "Name": "Name of Detection Rule",
    "Description": "Description of Detection Rule",
    "Severity": "Medium",
    "ObjectType": "DetectionRule",
    "OperatingSystems": [
        {
            "Name": "Windows",
        }
    ],
    "Plugin": {
        "Name": "OpticsDetector"
    },
    "Product": {
        "Name": "CylanceOPTICS"
    },
    "SchemaVersion": 1,
    "States": [
        {
            "Name": "MaliciousApp",
            "Scope": "Global",
            "Function": "Function",
            "FieldOperators": {
                "Function": {
                    "Type": "EqualsAny",
                    "Operands": [
                        {
                            "Source": "LiteralSet",
                            "Data": "badapp.exe"
                        }
                    ],
                    "OperandType": "string",
                    "Options": {
                        "IgnoreCase": true
                    }
                }
            },
            "Actions": [
                {
                    "Type": "AOI",
                    "ItemName": "InstigatingProcess",
                }
            ]
        }
    ]
}
```
"Position": "PostActivation"
}
],
"Filters": [
{
"Type": "Event",
"Data": {
"Category": "Process",
"SubCategory": "",
"Type": "*"
}
}
],
"Tags": [
"CylanceOPTICS"
]
}

Response:

200 OK

{
"valid": true,
"warnings": [],
"errors": []
}

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resources page requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

Request JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The description for the Detection Rule.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the Detection Rule.</td>
</tr>
<tr>
<td>ObjectType</td>
<td>The type of object defined in this rule.</td>
</tr>
<tr>
<td></td>
<td>- DetectionRule</td>
</tr>
<tr>
<td></td>
<td>- ResponseRule</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OperatingSystems</td>
<td>The affected operating systems.</td>
</tr>
<tr>
<td>Name:</td>
<td>The name of the type of operating system (like Windows, macOS, or Linux).</td>
</tr>
<tr>
<td>Plugin</td>
<td>The CylanceOPTICS plugin associated with the Detection Rule.</td>
</tr>
<tr>
<td>Product</td>
<td>The name of the product associated with the Detection Rule.</td>
</tr>
<tr>
<td>SchemaVersion</td>
<td>The version of the schema.</td>
</tr>
<tr>
<td>Severity</td>
<td>The severity assigned to the Detection Rule. Possible values are: High, Medium, Low, Informational</td>
</tr>
<tr>
<td>States</td>
<td>The list of all available states. If no paths are specified, the states are transitioned in the order they are specified.</td>
</tr>
<tr>
<td></td>
<td>See the <a href="#">&quot;Custom Rule Guide&quot; on page 331</a> for information about States.</td>
</tr>
<tr>
<td>Tags</td>
<td>A list of tags associated with the Detection Rule.</td>
</tr>
</tbody>
</table>

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>errors</td>
<td>A list of error messages that will prevent the Detection Rule from validating and operating correctly.</td>
</tr>
<tr>
<td>valid</td>
<td>Returns &quot;true&quot; if the Detection Rule passes validation. Returns &quot;false&quot; if the Detection Rule does not pass validation.</td>
</tr>
<tr>
<td>warnings</td>
<td>A list of warning message strings that may impact the performance or validity of the Detection Rule.</td>
</tr>
</tbody>
</table>

Create Detection Rule

Allows a caller to create a new Detection Rule by sending the native JSON structure of a Detection Rule.

Service Endpoint:

- /rules/v2

Example: https://protectapi.cylance.com/rules/v2
Method:
- HTTP/1.1 POST

Request Headers:
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsrule:create scope encoded

Request:

```json
{
    "Name": "Name of Detection Rule",
    "Description": "Description of Detection Rule",
    "Severity": "Medium",
    "ObjectType": "DetectionRule",
    "OperatingSystems": [
        {
            "Name": "Windows"
        }]
    }
    "Plugin": {
        "Name": "OpticsDetector"
    },
    "Product": {
        "Name": "CylanceOPTICS"
    },
    "SchemaVersion": 1,
    "States": [
        {
            "Name": "MaliciousApp",
            "Scope": "Global",
            "Function": "Function",
            "FieldOperators": {
                "Function": {
                    "Type": "EqualsAny",
                    "Operands": [
                        {
                            "Source": "LiteralSet",
                            "Data": "badapp.exe"
                        }
                    ],
                    "OperandType": "string",
                    "Options": {
                        "IgnoreCase": true
                    }
                }
            },
            "Actions": [
                {
                }
            ]
        }
    ]
}
```
"Type": "AOI",
"ItemName": "InstigatingProcess",
"Position": "PostActivation"
]
,"Filters": [
{
"Type": "Event",
"Data": {
"Category": "Process",
"SubCategory": "",
"Type": "*"
}
}
],
"Tags": [
"CylanceOPTICS"
]}

Response:

200 OK

{"Name": "Name of Detection Rule",
"Description": "Description of Detection Rule",
"Severity": "Medium",
"ObjectType": "DetectionRule",
"OperatingSystems": [
{
"Name": "Windows"
}
],
"Plugin": {
"Name": "OpticsDetector"
},
"Product": {
"Name": "CylanceOPTICS"
},
"SchemaVersion": 1,
"States": [
{
"Name": "MaliciousApp",
"Scope": "Global",
"Function": "Function",
"FieldOperators": {
"Function": {
"Type": "EqualsAny",
"Operands": [
"CylanceUser API Guide, v2.0 rev24, February 2020 | 223
"
400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resources page requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

The response JSON schema contains the entirety of the Detection Rule Logic. This schema is further explained in "Custom Rule Guide" on page 331.
Note: The "id" and "version" fields are automatically populated when the request is submitted.

**Request JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The description for the Detection Rule.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the Detection Rule.</td>
</tr>
<tr>
<td>ObjectType</td>
<td>The type of object defined in this rule.</td>
</tr>
<tr>
<td></td>
<td>- DetectionRule</td>
</tr>
<tr>
<td></td>
<td>- ResponseRule</td>
</tr>
<tr>
<td>OperatingSystems</td>
<td>The affected operating systems.</td>
</tr>
<tr>
<td></td>
<td>- Name: The name of the type of operating system (like Windows, macOS, or Linux).</td>
</tr>
<tr>
<td>Plugin</td>
<td>The CylanceOPTICS plugin associated with the Detection Rule.</td>
</tr>
<tr>
<td>Product</td>
<td>The name of the product associated with the Detection Rule.</td>
</tr>
<tr>
<td>SchemaVersion</td>
<td>The version of the schema.</td>
</tr>
<tr>
<td>Severity</td>
<td>The severity assigned to the Detection Rule. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>- High</td>
</tr>
<tr>
<td></td>
<td>- Medium</td>
</tr>
<tr>
<td></td>
<td>- Low</td>
</tr>
<tr>
<td></td>
<td>- Informational</td>
</tr>
<tr>
<td>States</td>
<td>The list of all available states. If no paths are specified, the states are transitioned in the order they are specified.</td>
</tr>
<tr>
<td></td>
<td>See the &quot;Custom Rule Guide&quot; on page 331 for information about States.</td>
</tr>
<tr>
<td>Tags</td>
<td>A list of tags associated with the Detection Rule.</td>
</tr>
</tbody>
</table>

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The description for the Detection Rule.</td>
</tr>
<tr>
<td>Id</td>
<td>The unique identifier for the Detection Rule.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the Detection Rule.</td>
</tr>
<tr>
<td>ObjectType</td>
<td>The type of object defined in this rule.</td>
</tr>
<tr>
<td></td>
<td>- DetectionRule</td>
</tr>
<tr>
<td></td>
<td>- ResponseRule</td>
</tr>
<tr>
<td>OperatingSystems</td>
<td>The affected operating systems.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the type of operating system (like Windows, macOS, or Linux).</td>
</tr>
<tr>
<td>Plugin</td>
<td>The CylanceOPTICS plugin associated with the Detection Rule.</td>
</tr>
<tr>
<td>Product</td>
<td>The name of the product associated with the Detection Rule.</td>
</tr>
<tr>
<td>RuleSourceGrouping</td>
<td>The classification or designator for the rule source. For example: Optics.</td>
</tr>
<tr>
<td>SchemaVersion</td>
<td>The version of the schema.</td>
</tr>
<tr>
<td>Severity</td>
<td>The severity assigned to the Detection Rule. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>- High</td>
</tr>
<tr>
<td></td>
<td>- Medium</td>
</tr>
<tr>
<td></td>
<td>- Low</td>
</tr>
<tr>
<td></td>
<td>- Informational</td>
</tr>
<tr>
<td>States</td>
<td>The list of all available states. If no paths are specified, the states are</td>
</tr>
<tr>
<td></td>
<td>transitioned in the order they are specified.</td>
</tr>
<tr>
<td></td>
<td>See the <a href="#">&quot;Custom Rule Guide&quot; on page 331</a> for information about States.</td>
</tr>
<tr>
<td>Tags</td>
<td>A list of tags associated with the Detection Rule.</td>
</tr>
<tr>
<td>Version</td>
<td>The version of the Detection Rule.</td>
</tr>
</tbody>
</table>

**Update Detection Rule**

Update a Detection Rule by sending a new JSON structure.

**Service Endpoint:**

- /rules/v2/{rule_id}

**Example:** https://protectapi.cylance.com/rules/v2/c407f28a-3805-4014-b32c-0c2553ac1e10

**Method:**

- HTTP/1.1 PUT

**Request Headers:**

- Accept: application/json

- Authorization: Bearer <JWT Token returned by Auth API> with the opticsrule:update scope encoded

**Request:**
{  
"Name": "Name of Detection Rule",
"Description": "Description of Detection Rule",
"Severity": "High",
"ObjectType": "DetectionRule",
"OperatingSystems": [  
{  
"Name": "Windows"  
}  
],
"Plugin": {  
"Name": "OpticsDetector"  
},
"Product": {  
"Name": "CylanceOPTICS"  
},
"SchemaVersion": 1,
"States": [  
{  
"Name": "MaliciousApp",
"Scope": "Global",
"Function": "Function",
"FieldOperators": {  
"Function": {  
"Type": "EqualsAny",
"Operands": [  
{  
"Source": "LiteralSet",
"Data": "badapp.exe"  
}  
],
"OperandType": "string",
"Options": {  
"IgnoreCase": true  
}  
}  
},
"Actions": [  
{  
"Type": "AOI",
"ItemName": "InstigatingProcess",
"Position": "PostActivation"  
}  
],
"Filters": [  
{  
"Type": "Event",
"Data": {  
"Category": "Process",
"SubCategory": "",
"Type": "*"  
}  
}  
]  
}  
}
Response:

202 OK - Rule created successfully

{
  "Name": "Name of Detection Rule",
  "Description": "Description of Detection Rule",
  "Severity": "High",
  "ObjectType": "DetectionRule",
  "OperatingSystems": [
    {
      "Name": "Windows"
    }
  ],
  "Plugin": {
    "Name": "OpticsDetector"
  },
  "Product": {
    "Name": "CylanceOPTICS"
  },
  "SchemaVersion": 1,
  "States": [
    {
      "Name": "MaliciousApp",
      "Scope": "Global",
      "Function": "Function",
      "FieldOperators": {
        "Function": {
          "Type": "EqualsAny",
          "Operands": [
            {
              "Source": "LiteralSet",
              "Data": "badapp.exe"
            }
          ],
          "OperandType": "string",
          "Options": {
            "IgnoreCase": true
          }
        }
      },
      "Actions": [
        {
        }
      ]
    }
  ]
}
"Type": "AOI",
"ItemName": "InstigatingProcess",
"Position": "PostActivation"
],
"Filters": [
{
"Type": "Event",
"Data": {
"Category": "Process",
"SubCategory": "",
"Type": "*"
}
}
]
},
"Tags": [
  "CylanceOPTICS"
],
"RuleSourceGrouping": "Custom Rule",
"Id": "008ece50-49af-472a-b0d8-3c3700883738",
"Version": 2
}

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resources page requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

The response JSON schema contains the entirety of the Detection Rule Logic. This schema is further explained in "Custom Rule Guide" on page 331.

**Note:** The "id" and "version" fields are automatically populated when the request is submitted.

**Request JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The description for the Detection Rule.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the Detection Rule.</td>
</tr>
<tr>
<td>ObjectType</td>
<td>The type of object defined in this rule.</td>
</tr>
<tr>
<td></td>
<td>- DetectionRule</td>
</tr>
<tr>
<td></td>
<td>- ResponseRule</td>
</tr>
</tbody>
</table>

Cylance User API Guide, v2.0 rev24, February 2020 | 229
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OperatingSystems</td>
<td>The affected operating systems.</td>
</tr>
<tr>
<td></td>
<td>- Name: The name of the type of operating system (like Windows, macOS, or Linux).</td>
</tr>
<tr>
<td>Plugin</td>
<td>The CylanceOPTICS plugin associated with the Detection Rule.</td>
</tr>
<tr>
<td>Product</td>
<td>The name of the product associated with the Detection Rule.</td>
</tr>
<tr>
<td>SchemaVersion</td>
<td>The version of the schema.</td>
</tr>
<tr>
<td>Severity</td>
<td>The severity assigned to the Detection Rule. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>- High</td>
</tr>
<tr>
<td></td>
<td>- Medium</td>
</tr>
<tr>
<td></td>
<td>- Low</td>
</tr>
<tr>
<td></td>
<td>- Informational</td>
</tr>
<tr>
<td>States</td>
<td>The list of all available states. If no paths are specified, the states are transitioned in the order they are specified.</td>
</tr>
<tr>
<td></td>
<td>See the &quot;Custom Rule Guide&quot; on page 331 for information about States.</td>
</tr>
<tr>
<td>Tags</td>
<td>A list of tags associated with the Detection Rule.</td>
</tr>
</tbody>
</table>

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The description for the Detection Rule.</td>
</tr>
<tr>
<td>Id</td>
<td>The unique identifier for the Detection Rule.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the Detection Rule.</td>
</tr>
<tr>
<td>ObjectType</td>
<td>The type of object defined in this rule.</td>
</tr>
<tr>
<td></td>
<td>- DetectionRule</td>
</tr>
<tr>
<td></td>
<td>- ResponseRule</td>
</tr>
<tr>
<td>OperatingSystems</td>
<td>The affected operating systems.</td>
</tr>
<tr>
<td></td>
<td>- Name: The name of the type of operating system (like Windows, macOS, or Linux).</td>
</tr>
<tr>
<td>Plugin</td>
<td>The CylanceOPTICS plugin associated with the Detection Rule.</td>
</tr>
<tr>
<td>Product</td>
<td>The name of the product associated with the Detection Rule.</td>
</tr>
<tr>
<td>RuleSourceGrouping</td>
<td>The classification or designator for the rule source. For example: Optics.</td>
</tr>
<tr>
<td>SchemaVersion</td>
<td>The version of the schema.</td>
</tr>
<tr>
<td>Severity</td>
<td>The severity assigned to the Detection Rule. Possible values are:</td>
</tr>
</tbody>
</table>
### Field Name | Description
--- | ---
- High  
- Medium  
- Low  
- Informational

**States**  
The list of all available states. If no paths are specified, the states are transitioned in the order they are specified.  
See the ["Custom Rule Guide" on page 331](#) for information about States.

**Tags**  
A list of tags associated with the Detection Rule.

**Version**  
The version of the Detection Rule.

---

### Deactivate or Delete Detection Rule

"Soft delete" a Detection Rule and remove it from the Detection Rule Sets.

**Service Endpoint:**

- `/rules/v2/{rule_id}/deactivate`

**Example:**  
https://protectapi.cylance.com/rules/v2/c407f28a-3805-4014-b32c-0c2553ac1e10/deactivate

**Method:**

- HTTP/1.1 POST

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsrule:update scope encoded

**Request**

- None

**Response**

- 200 OK
- 400 Bad Request - Malformed request.
- 401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
- 403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
404 Not Found - The detection resources page requested doesn't exist.
500 Internal Server Error - An unforeseeable error has occurred.
503 Service Unavailable - Unable to respond at this time, please try again later.

**Note:** Detection Rule Sets are not automatically communicated to all endpoints when updates to Detection Rules are made. To ensure that the latest logic is applied to endpoints in the quickest manner, re-save any affected Detection Rule Sets (either via the UI or API).

## Get Detection Rule Natural Language Representation

Retrieve the "natural language" representation of a rule. This process converts the Detection Rule logic into a series of 'AND's, 'OR's, and 'NOT's to describe what the Detection Rule looks for. The underlying logic extracts from the JWT specified as the Bearer value in the Authorization request header the tenant's unique identifier to associated the Detection Rule resource with.

**Service Endpoint:**

- `/rules/v2/{rule_id}/natlang`

**Example:** https://protectapi.cylance.com/rules/v2/c407f28a-3805-4014-b32c-0c2553ac1e10/natlang

**Method:**

- HTTP/1.1 GET

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsrule:read scope encoded

**Request**

- None

**Response**

- 200 OK

```json
{
    "Name": "Name of Detection Rule",
    "Paths": [
        [
```
Get Detection Rule Counts

Retrieve counts of how many devices, Detection Rule Sets, and policies that have a particular Detection Rule applied.

**Service Endpoint:**

- `/rules/v2/{rule_id}/counts`

**Example:** https://protectapi.cylance.com/rules/v2/c407f28a-3805-4014-b32c-0c2553ac1e10/counts

**Method:**

- HTTP/1.1 GET

**Request Headers:**

- Accept: application/json
- Authorization: Bearer `<JWT Token returned by Auth API>` with the opticsrule:read scope encoded

**Request**

- None

**Response**

- 200 OK

```json
{
    "DeviceCount": 0,
    "RulesetCount": 0,
    "PolicyCount": 0
}
```
- 400 Bad Request - Malformed request.
- 401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
- 403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
- 404 Not Found - The detection resources page requested doesn't exist.
- 500 Internal Server Error - An unforeseeable error has occurred.
- 503 Service Unavailable - Unable to respond at this time, please try again later.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeviceCount</td>
<td>The number of devices that have the requested Detection Rule applied.</td>
</tr>
<tr>
<td>PolicyCount</td>
<td>The number of Device Policies that have the requested Detection Rule applied.</td>
</tr>
<tr>
<td>RulesetCount</td>
<td>The number of Detection Rule Sets that have the requested Detection Rule enabled.</td>
</tr>
</tbody>
</table>
Detection Rule Sets
The CylanceOPTICS Detection Rule Set API allows users to create a set of rules and apply that set to Device Policies.

The CylanceOPTICS Detection Rule Set API includes:

- Getting content for a Detection Rule Set
- Getting a list of Detection Rule Sets
- Creating a Detection Rule Set
- Retrieving a Default Detection Rule Set (retrieving a default template)
- Updating a Detection Rule Set
- Deleting a Detection Rule Set
- Deleting multiple Detection Rule Sets
- Getting a list of Detection Rule Sets as a CSV file

### Get Detection Rule Set List

Retrieve a list of Detection Rule Sets available in a tenant.

**Service Endpoint:**

- `/rulesets/v2?page=m&page_size=n`

Append the following optional query string parameters:

- page: The page number to request
- page_size: The number of device records to retrieve per page

For example, to return the first page with 100 devices:

```
https://protectapi.cylance.com/rulesets/v2?page=1&page_size=100
```

**Method:**

- HTTP/1.1 GET

**Request Headers:**

- Accept: application/json
- Authorization: Bearer &lt;JWT Token returned by Auth API&gt; with the opticsruleset:list scope encoded

**Request**

Append the following parameters to the request to sort and filter the results:
- **description**: Case-insensitive query parameter to filter or sort by the Description field.
  
  **Example**: https://protectapi.cylance.com/rulesets/v2?page=1&page_size=100&description=test
  
  This example returns the first page, up to 100 items, with "test" contained in the Description.

- **last_modified**: Case-insensitive query parameter to filter or sort by the Last Modified field. An example of the date/time format: 2019-04-10T21:39:54Z. Partial information will return matching results. For example, if 2019-04-10, 21:39:54, or 2019-04-10T21 is used, 2019-04-10T21:39:54Z will return, along with any other matching results.

  **Example**: https://protectapi.cylance.com/rulesets/v2?page=1&page_size=100&last_modified=2019-04-10
  
  This example returns the first page, up to 100 items, with a last modified date of April 10, 2019.

  This parameter can also include the time. For example, 2019-04-10T21:39:54Z.

  **Note**: Entering partial information will return all Detection Rule Sets with a Last Modified date that contains that string. For example, using 2018 returns all Detection Rule Sets modified in the year 2018.

- **modified_by.id**: Case-insensitive query parameter to filter or sort by a user's unique ID.

  **Example**: https://protectapi.cylance.com/rulesets/v2?page=1&page_size=100&modified_by.id=8e6e9ef6-14fe-4eb4-8895-44d62b532b64
  
  This example returns the first page, up to 100 items, with the specific user ID.

  **Note**: Entering partial information will return all Detection Rule Sets with a user ID that contains that string. For example, using 8 returns all user ID's that start with and include that value.

- **modified_by.login**: Case-insensitive query parameter to filter or sort by a user's email address.

  **Example**: https://protectapi.cylance.com/rulesets/v2?page=1&page_size=100&modified_by.login=username@email.com
  
  This example returns the first page, up to 100 items, with the specific email address.

  **Note**: Entering partial information will return all Detection Rule Sets with a user's email address that contains that string. For example, using "j" will return "jdoe@email.com" and "ajames@email.com".

- **device_count**: Filter or sort the list by the number of applied devices.
Example: https://protectapi.cylance.com/rulesets/v2?page=1&page_size=100&device_count=10

This example returns the first page, up to 100 items, with a Device Count that starts with "10," which includes 100 and 1010.

- **sort**: Sort by field (adding '-' in front of the value denotes descending order).

  Example: https://protectapi.cylance.com/rulesets/v2?page=1&page_size=100&device_count=10&sort=policies

  This example returns the first page, up to 100 items, with a Device Count that starts with "10," and sorted by policy ID (in ascending order). Placing a '-' in front of the sort value sorts in descending order - "&sort=-policies".

Response

200 OK

```json
{
  "page_size": "1",
  "total_pages": "10",
  "page_items": [
  {
    "name": "Test Rule Set",
    "description": "Test Detection Rule Set",
    "notification_message": "",
    "id": "008ece50-49af-472a-b0d8-3c3700883738",
    "last_modified": "2018-07-26T01:20:07.596Z",
    "modified_by": {
      "id": "a2c0ac7a-a63d-4583-b646-ae10db9c9769",
      "login": "testuser@email.com"
    },
    "policies": [
      "d5c6d6a3-0599-4fb5-96bc-0f0dc7eacb6e"
    ],
    "device_count": 1,
    "category": "Custom"
  }
  ],
  "total_number_of_items": 1,
  "page_number": 1
}
```

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resources page requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.
Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>page_size</td>
<td>The number of items on the page.</td>
</tr>
<tr>
<td>total_pages</td>
<td>The total number of pages of this size.</td>
</tr>
<tr>
<td>total_number_of_items</td>
<td>The total number of Detection Rules in the tenant.</td>
</tr>
<tr>
<td>page_number</td>
<td>The current page number of results.</td>
</tr>
<tr>
<td>page_items</td>
<td>A list of Exception objects that are available in the tenant that will contain the following fields.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the Detection Rule.</td>
</tr>
<tr>
<td>description</td>
<td>The description of the Detection Rule.</td>
</tr>
<tr>
<td>id</td>
<td>The unique ID of the Detection Rule.</td>
</tr>
<tr>
<td>last_modified</td>
<td>The timestamp (in UTC) of the last time that the Detection Rule was modified.</td>
</tr>
<tr>
<td>modified_by</td>
<td>An object detailing the last user to modify the Detection Rule. It includes the following fields:</td>
</tr>
<tr>
<td></td>
<td>id: The unique ID of the user who modified the Detection Rule.</td>
</tr>
<tr>
<td></td>
<td>login: The email address of the user who modified the Detection Rule.</td>
</tr>
<tr>
<td>policies</td>
<td>A list of policy IDs that a Detection Rule Set is applied to.</td>
</tr>
<tr>
<td>device_count</td>
<td>The number of devices that have the Detection Rule applied.</td>
</tr>
<tr>
<td>category</td>
<td>The category of rule grouping that the Detection Rule belongs to. Possible values include:</td>
</tr>
<tr>
<td></td>
<td>Custom: Custom rules that users have uploaded to a tenant.</td>
</tr>
<tr>
<td></td>
<td>Cylance Rules: Cylance-official rules.</td>
</tr>
<tr>
<td></td>
<td>Cylance Experimental: Cylance rules that are deemed to be experimental in their nature.</td>
</tr>
</tbody>
</table>

Get Detection Rule Set CSV List

Retrieve a CSV where every line represents a Detection Rule Set available in a tenant.

Service Endpoint:
- /rulesets/v2/csv

For example: https://protectapi.cylance.com/rulesets/v2/csv

Method:
HTTP/1.1 GET

Request Headers:
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsruleset:list scope encoded

Request
None

Response
200 OK - A proper response displays the requested detection information in CSV format.

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such resource found.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

Response CSV Contains the Following Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>The unique ID of the Exception.</td>
</tr>
<tr>
<td>Last Modified</td>
<td>The timestamp (in UTC) of the last time the Detection Rule Set was modified.</td>
</tr>
<tr>
<td>Modified By</td>
<td>The email address of the user who last modified the Detection Rule Set.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the Detection Rule Set.</td>
</tr>
<tr>
<td>Description</td>
<td>The description of the Detection Rule Set.</td>
</tr>
<tr>
<td>Notification</td>
<td>The Notification Message to display on a device if the Detection Rule triggers.</td>
</tr>
<tr>
<td>Category</td>
<td>The category of the Detection Rule Set.</td>
</tr>
<tr>
<td>Device Count</td>
<td>The number of devices that have the Detection Rule Set applied.</td>
</tr>
</tbody>
</table>

Get Detection Rule Set

Retrieve the content of a Detection Rule Set, including Detection Rules, Response Actions, Detection Exceptions, Package Playbooks, and the Policies where the Detection Rule Set is applied.
Service Endpoint:

- /rulesets/v2/{ruleset_id}

For example: https://protectapi.cylance.com/rulesets/v2/c407f28a-3805-4014-b32c-0c2553ac1e17

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsruleset:read scope encoded

Request

None

Response

200 OK

```json
{
  "name": "Test Rule Set",
  "description": "Test Detection Rule Set",
  "notification_message": "",
  "category": "Custom",
  "id": "008ece50-49af-472a-b0d8-3c3700883738",
  "last_modified": "2018-07-26T01:20:07.596Z",
  "modified_by": {
    "id": "a2c0ac7a-a63d-4583-b646-ae10db9c9769",
    "login": "testuser@email.com"
  },
  "rules": [
    {
      "detection_rule_id": "008ece50-49af-472a-b0d8-3c3700883738",
      "detection_rule_version": 1,
      "detection_name": "Gatekeeper Bypass (MITRE)",
      "detection_description": "Detects on usage to bypass Gatekeeper",
      "category": "Cylance Rules",
      "severity": "Low",
      "operating_systems": [
        {
          "Name": "macOS"
        }
      ],
      "date_added": "2018-11-20T17:58:49Z",
      "enabled": false,
    }
  ]
}
```
"notification_enabled": ,
"responses": [
{
"template_id": "9686d8e-1b1d-45a9-977a-cf86f1063b15",
"response_id": "c6a26a8b-edce-4a68-8e18-4d16df74e455",
"response_rule_version": 1,
"description": "DisplayNotification",
"value": {},
"enabled": false,
"created": "2018-11-20T17:58:49Z"
}
],
"exceptions": [
{
"exception_id": "0f12a426-a956-4f4e-a698-df732b1b296",
"enabled": false,
"name": "Test Exception"
}
],
"playbooks": []
}
)

400 Bad Request - Malformed request.
401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
404 Not Found - The detection resources page requested doesn't exist.
500 Internal Server Error - An unforeseeable error has occurred.
503 Service Unavailable - Unable to respond at this time, please try again later

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the Detection Rule Set.</td>
</tr>
<tr>
<td>description</td>
<td>The description of the Detection Rule Set.</td>
</tr>
<tr>
<td>notification_message</td>
<td>The message to display on the endpoint when a Detection Rule is triggered.</td>
</tr>
<tr>
<td>id</td>
<td>The unique ID of the Detection Rule Set.</td>
</tr>
<tr>
<td>last_modified</td>
<td>The timestamp (in UTC) of the last time that the Detection Rule Set was modified.</td>
</tr>
<tr>
<td>modified_</td>
<td>An object detailing the last user to modify the Detection Rule. It includes the</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| by         | following fields:  
  - id: The unique ID of the user who modified the Detection Rule.  
  - login: The email address of the user who modified the Detection Rule. |
| detection_rule_id | The unique ID of the Detection Rule. |
| detection_rule_version | The version of the Detection Rule. |
| detection_name | The name of the Detection Rule. |
| detection_description | The description of the Detection Rule Set. |
| category | The category of the Detection Rule. |
| severity | The severity assigned to the Detection Rule. Possible values are:  
  - High  
  - Medium  
  - Low  
  - Informational |
| operating_systems | An object detailing the operating systems to which the Detection Rule can be applied. It will include the "name" field. This can consist of:  
  - "Windows"  
  - "MacOS" |
| date_added | The timestamp (in UTC) when the Detection Rule was added to the tenant. |
| enabled | Determines whether or not a Detection Rule is enabled in the Detection Rule Set. When viewing the content of a Detection Rule Set, this should always be set to 'true'. |
| notification_enabled | Determines whether or not the message defined in the 'notification_message' field should display on the device when the Detection Rule is triggered.  
  **Note:** To enable Display Desktop Notification on Device using the API, set notification_enabled and DisplayDesktopNotification to "true". To disable, set both to "false". The DisplayDesktopNotification setting enables or disables the feature. The notification_enabled setting affects the Display Desktop Notification on Device checkbox in the Console as enabled (checked) or disabled (unchecked). |
<p>| responses | A list of response objects for each Response Action enabled for a particular Detection Rule. Each object will include the following fields: |</p>
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>template_id</td>
<td>The ID of the response template to use (this is provided by Cylance).</td>
</tr>
<tr>
<td>response_rule_id</td>
<td>The ID of the response rule to enable (these are provided by Cylance).</td>
</tr>
<tr>
<td>response_rule_version</td>
<td>The version of the response rule to enable (this is provided by Cylance).</td>
</tr>
<tr>
<td>description</td>
<td>The description / name of the response rule.</td>
</tr>
<tr>
<td>value</td>
<td>A currently unused field.</td>
</tr>
<tr>
<td>enabled</td>
<td>This will always be 'true' when viewing a Detection Rule Set.</td>
</tr>
<tr>
<td>created</td>
<td>The date that the Response Rule was added to the tenant.</td>
</tr>
</tbody>
</table>

**exceptons**

A list of Exception Rule objects that should be applied to the Detection Rule. Each object will include the following fields:

- exception_id: The unique ID of the Exception Rule.
- enabled: This will always be 'true' when viewing a Detection Rule Set.
- name: The name of the Exception Rule.

**playbooks**

A list of Package Playbook unique IDs that will be executed when the Detection Rule is triggered on the device.

---

**Create Detection Rule Set**

Create a new Detection Rule Set. Detection Rule Sets can require a large number of fields and unique IDs to function properly.

**Tip:** It is recommended to make a GET request to '/rulesets/v2/default' to obtain a properly formatted template prior to submitting a POST request described below.

**Service Endpoint:**

- /rulesets/v2

For example: https://protectapi.cylance.com/rulesets/v2

**Method:**

- HTTP/1.1 POST

**Request Headers:**

- Accept: application/json

- Authorization: Bearer <JWT Token returned by Auth API> with the opticsruleset:create scope encoded

**Request**
{ "name": "Test Rule Set", "description": "Test Detection Rule Set", "notification_message": "", "category": "Custom", "rules": [ { "detection_rule_id": "008ece50-49af-472a-b0d8-3c3700883738", "detection_rule_version": 1, "detection_name": "Gatekeeper Bypass (MITRE)" , "detection_description": "Detects on usage to bypass Gatekeeper", "category": "Custom", "severity": "Low", "operating_systems": [ { "Name": "macOS" } ], "date_added": "2018-11-20T17:58:49Z", "enabled": false, "notification_enabled": false, "responses": [ { "template_id": "9686d82e-1b1d-45a9-977a-cf86f1063b15", "response_id": "c6a26a8b-edce-4a68-8e18-4d16df74e455", "response_rule_version": 1, "description": "DisplayNotification", "value": {}, "enabled": false, "created": "2018-11-20T17:58:49Z" } ] }, "exceptions": [ { "exception_id": "", "enabled": false, "name": "" } ], "playbooks": [ "" ] } }

Response

202 OK - Ruleset created

{
400 Bad Request - Malformed request.
401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resources page requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the Detection Rule Set.</td>
</tr>
<tr>
<td>description</td>
<td>The description of the Detection Rule Set.</td>
</tr>
<tr>
<td>notification_message</td>
<td>The message to display on the endpoint when a Detection Rule is triggered.</td>
</tr>
<tr>
<td>id</td>
<td>The unique ID of the Detection Rule Set.</td>
</tr>
<tr>
<td>last_modified</td>
<td>The timestamp (in UTC) of the last time that the Detection Rule Set was modified.</td>
</tr>
<tr>
<td>modified_by</td>
<td>An object detailing the last user to modify the Detection Rule. It includes the following fields:</td>
</tr>
<tr>
<td></td>
<td>- id: The unique ID of the user who modified the Detection Rule.</td>
</tr>
<tr>
<td></td>
<td>- login: The email address of the user who modified the Detection Rule.</td>
</tr>
<tr>
<td>detection_rule_id</td>
<td>The unique ID of the Detection Rule.</td>
</tr>
<tr>
<td>detection_rule_version</td>
<td>The version of the Detection Rule.</td>
</tr>
<tr>
<td>detection_name</td>
<td>The name of the Detection Rule.</td>
</tr>
<tr>
<td>detection_description</td>
<td>The description of the Detection Rule Set.</td>
</tr>
<tr>
<td>category</td>
<td>The category of the Detection Rule.</td>
</tr>
<tr>
<td>severity</td>
<td>The severity assigned to the Detection Rule. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>- High</td>
</tr>
<tr>
<td></td>
<td>- Medium</td>
</tr>
<tr>
<td></td>
<td>- Low</td>
</tr>
</tbody>
</table>

Cylance User API Guide, v2.0 rev24, February 2020 | 247
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| operating_systems  | An object detailing the operating systems to which the Detection Rule can be applied. It will include the "name" field. This can consist of:  
  - "Windows"  
  - "MacOS"                                                                                                                                       |
| date_added         | The timestamp (in UTC) when the Detection Rule was added to the tenant.                                                                                                                                    |
| enabled            | Determines whether or not a Detection Rule is enabled in the Detection Rule Set. When viewing the content of a Detection Rule Set, this should always be set to 'true'.                                             |
| notification_enabled | Determines whether or not the message defined in the 'notification_message' field should display on the device when the Detection Rule is triggered.  
  **Note:** To enable Display Desktop Notification on Device using the API, set notification_enabled and DisplayDesktopNotification to "true". To disable, set both to "false". The DisplayDesktopNotification setting enables or disables the feature. The notification_enabled setting affects the Display Desktop Notification on Device checkbox in the Console as enabled (checked) or disabled (unchecked). |
| responses          | A list of response objects for each Response Action enabled for a particular Detection Rule. Each object will include the following fields:  
  - template_id: The ID of the response template to use (this is provided by Cylance).  
  - response_rule_id: The ID of the response rule to enable (these are provided by Cylance).  
  - response_rule_version: The version of the response rule to enable (this is provided by Cylance).  
  - description: The description / name of the response rule.  
  - value: A currently unused field.  
  - enabled: This will always be 'true' when viewing a Detection Rule Set.  
  - created: The date that the Response Rule was added to the tenant.                                                                                     |
| exceptions         | A list of Exception Rule objects that should be applied to the Detection Rule. Each object will include the following fields:  
  - exception_id: The unique ID of the Exception Rule.  
  - enabled: This will always be 'true' when viewing a Detection Rule Set.  
  - name: The name of the Exception Rule.                                                                                                               |
| playbooks          | A list of Package Playbook unique IDs that will be executed when the Detection Rule is triggered on the device.                                                                                             |
Retrieve Default Detection Rule Set

Retrieve a properly formatted default Detection Rule Set template that includes all Detection Rules, Exceptions, Playbooks, and Response Actions available in a tenant. The output of this request can be modified and submitted as a POST request to 'ruleset/v2' to create a new Detection Rule Set.

**Service Endpoint:**
- /rulesets/v2/default

**For example:** https://protectapi.cylance.com/rulesets/v2/default

**Method:**
- HTTP/1.1 GET

**Request Headers:**
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsruleset:read scope encoded

**Request**

None

**Response**

200 OK

```json
{
  "name": "",
  "description": "",
  "notification_message": "",
  "category": "Custom",
  "rules": [
    {
      "detection_rule_id": "998ece50-49af-472a-b0d8-3c3700883736",
      "detection_rule_version": 1,
      "detection_name": "Gatekeeper Bypass (MITRE)",
      "detection_description": "Detects on usage of xattr or spctl to bypass Gatekeeper, by a non-root user (MITRE1144)",
      "category": "Cylance MITRE ATT&CK Rules",
      "severity": "Low",
      "operating_systems": [
        {
          "Name": "macOS"
        }
      ],
      "date_added": "2018-11-20T17:58:49Z"
    }
  ]
}```
"enabled": false,
"notification_enabled": false,
"responses": [
  {
    "template_id": "9986d82e-1b1d-45a9-977a-cf86f1063b14",
    "response_id": "95947b5c-71ce-4a7e-a5e0-df5043402b5c",
    "response_rule_version": 1,
    "description": "DisplayDesktopNotification",
    "value": {},
    "enabled": false,
    "created": "2018-11-20T17:58:49Z"
  }
],
"exceptions": [
  {
    "exception_id": "9f12a426-a956-4f4e-a698-df732b1b295",
    "enabled": false,
    "name": "AO Exception"
  }
],
"playbooks": []
]
}

400 Bad Request - Malformed request.
401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
404 Not Found - The detection resources page requested doesn't exist.
500 Internal Server Error - An unforeseeable error has occurred.
503 Service Unavailable - Unable to respond at this time, please try again later

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the Detection Rule Set.</td>
</tr>
<tr>
<td>description</td>
<td>The description of the Detection Rule Set.</td>
</tr>
<tr>
<td>notification_message</td>
<td>The message to display on the endpoint when a Detection Rule is triggered.</td>
</tr>
<tr>
<td>detection_</td>
<td>The unique ID of the Detection Rule.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>rule_id</td>
<td></td>
</tr>
<tr>
<td>detection_rule_version</td>
<td>The version of the Detection Rule.</td>
</tr>
<tr>
<td>detection_name</td>
<td>The name of the Detection Rule.</td>
</tr>
<tr>
<td>detection_description</td>
<td>The description of the Detection Rule Set.</td>
</tr>
<tr>
<td>category</td>
<td>The category of the Detection Rule.</td>
</tr>
<tr>
<td>severity</td>
<td>The severity assigned to the Detection Rule. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>■ High</td>
</tr>
<tr>
<td></td>
<td>■ Medium</td>
</tr>
<tr>
<td></td>
<td>■ Low</td>
</tr>
<tr>
<td></td>
<td>■ Informational</td>
</tr>
<tr>
<td>operating_systems</td>
<td>An object detailing the operating systems to which the Detection Rule can</td>
</tr>
<tr>
<td></td>
<td>be applied. It will include the &quot;name&quot; field. This can consist of:</td>
</tr>
<tr>
<td></td>
<td>■ &quot;Windows&quot;</td>
</tr>
<tr>
<td></td>
<td>■ &quot;MacOS&quot;</td>
</tr>
<tr>
<td>date_added</td>
<td>The timestamp (in UTC) when the Detection Rule was added to the tenant.</td>
</tr>
<tr>
<td>enabled</td>
<td>Determines whether or not a Detection Rule is enabled in the Detection</td>
</tr>
<tr>
<td></td>
<td>Rule Set. When viewing the content of a Detection Rule Set, this should</td>
</tr>
<tr>
<td></td>
<td>always be set to 'true'.</td>
</tr>
<tr>
<td>notification_enabled</td>
<td>Determines whether or not the message defined in the 'notification_</td>
</tr>
<tr>
<td></td>
<td>message' field should display on the device when the Detection Rule is</td>
</tr>
<tr>
<td></td>
<td>triggered.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> To enable Display Desktop Notification on Device using the API,</td>
</tr>
<tr>
<td></td>
<td>set notification_enabled and DisplayDesktopNotification to &quot;true&quot;. To</td>
</tr>
<tr>
<td></td>
<td>disable, set both to &quot;false&quot;. The DisplayDesktopNotification setting</td>
</tr>
<tr>
<td></td>
<td>enables or disables the feature. The notification_enabled setting affects</td>
</tr>
<tr>
<td></td>
<td>the Display Desktop Notification on Device checkbox in the Console as</td>
</tr>
<tr>
<td></td>
<td>enabled (checked) or disabled (unchecked).</td>
</tr>
<tr>
<td>responses</td>
<td>A list of response objects for each Response Action enabled for a particular</td>
</tr>
<tr>
<td></td>
<td>Detection Rule. Each object will include the following fields:</td>
</tr>
<tr>
<td></td>
<td>■ template_id: The ID of the response template to use (this is provided</td>
</tr>
<tr>
<td></td>
<td>by Cylance).</td>
</tr>
<tr>
<td></td>
<td>■ response_rule_id: The ID of the response rule to enable (these are</td>
</tr>
<tr>
<td></td>
<td>provided by Cylance).</td>
</tr>
<tr>
<td></td>
<td>■ response_rule_version: The version of the response rule to enable (this</td>
</tr>
<tr>
<td></td>
<td>is provided by Cylance).</td>
</tr>
<tr>
<td></td>
<td>■ description: The description / name of the response rule.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>value</td>
<td>A currently unused field.</td>
</tr>
<tr>
<td>enabled</td>
<td>This will always be 'true' when viewing a Detection Rule Set.</td>
</tr>
<tr>
<td>created</td>
<td>The date that the Response Rule was added to the tenant.</td>
</tr>
</tbody>
</table>

 exceptions
A list of Exception Rule objects that should be applied to the Detection Rule. Each object will include the following fields:
- exception_id: The unique ID of the Exception Rule.
- enabled: This will always be 'true' when viewing a Detection Rule Set.
- name: The name of the Exception Rule.

 playbooks
A list of Package Playbook unique IDs that will be executed when the Detection Rule is triggered on the device.

### Update Detection Rule Set

Update a Detection Rule Set by sending a new JSON structure.

**Service Endpoint:**
- `/rulesets/v2/{ruleset_id}`

For example: `https://protectapi.cylance.com/rulesets/v2/c407f28a-3805-4014-b32c-0c2553acle17`

**Method:**
- HTTP/1.1 PUT

**Request Headers:**
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsruleset: update scope encoded

**Request**

```json
{
    "name": "",
    "description": "",
    "notification_message": "",
    "category": "Custom",
    "rules": [
        {
            "detection_rule_id": "998ece50-49af-472a-b0d8-3c370083736",
            "detection_rule_version": 1,
        }
    ]
}
```
"detection_name": "Gatekeeper Bypass (MITRE)",
"detection_description": "Detects on usage of xattr or spctl to bypass Gatekeeper, by a non-root user (MITRE1144)",
"category": "Cylance MITRE ATT&CK Rules",
"severity": "High",
"operating_systems": [
  {
    "Name": "macOS"
  }
],
"date_added": "2018-11-20T17:58:49Z",
"enabled": false,
"notification_enabled": false,
"responses": [
  {
    "template_id": "9986d82e-1b1d-45a9-977a-cf86f1063b14",
    "response_id": "95947b5c-71ce-4a7e-a5e0-df5043402b5c",
    "response_rule_version": 1,
    "description": "DisplayDesktopNotification",
    "value": {},
    "enabled": false,
    "created": "2018-11-20T17:58:49Z"
  }
],
"exceptions": [
  {
    "exception_id": "9f12a426-a956-4f4e-a698-df732ba1b295",
    "enabled": false,
    "name": "AO Exception"
  }
],
"playbooks": []
}

Response

202 OK - Ruleset updated

{  
  "name": "",
  "description": "",
  "notification_message": "",
  "category": "Custom",
  "rules": [  
    {  
      "detection_rule_id": "998ece50-49af-472a-b0d8-3c3700883736",
      "detection_rule_version": 1,
      "detection_name": "Gatekeeper Bypass (MITRE)",
      "detection_description": "Detects on usage of xattr or spctl to bypass Gatekeeper, by a non-root user (MITRE1144)"
    }
  ]
}
"category": "Cylance MITRE ATT&CK Rules",
"severity": "High",
"operating_systems": [
  {
    "Name": "macOS"
  }
],
"date_added": "2018-11-20T17:58:49Z",
"enabled": false,
"notification_enabled": false,
"responses": [
  {
    "template_id": "9986d82e-1b1d-45a9-977a-cf86f1063b14",
    "response_id": "95947b5c-71ce-4a7e-a5e0-df5043402b5c",
    "response_rule_version": 1,
    "description": "DisplayDesktopNotification",
    "value": {},
    "enabled": false,
    "created": "2018-11-20T17:58:49Z"
  }
],
"exceptions": [
  {
    "exception_id": "9f12a426-a956-4f4e-a698-df732ba1b295",
    "enabled": false,
    "name": "AO Exception"
  }
],
"playbooks": []
}

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resources page requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the Detection Rule Set.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>description</td>
<td>The description of the Detection Rule Set.</td>
</tr>
<tr>
<td>notification_message</td>
<td>The message to display on the endpoint when a Detection Rule is triggered.</td>
</tr>
<tr>
<td>detection_rule_id</td>
<td>The unique ID of the Detection Rule.</td>
</tr>
<tr>
<td>detection_rule_version</td>
<td>The version of the Detection Rule.</td>
</tr>
<tr>
<td>detection_name</td>
<td>The name of the Detection Rule.</td>
</tr>
<tr>
<td>detection_description</td>
<td>The description of the Detection Rule Set.</td>
</tr>
<tr>
<td>category</td>
<td>The category of the Detection Rule.</td>
</tr>
<tr>
<td>severity</td>
<td>The severity assigned to the Detection Rule. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>- High</td>
</tr>
<tr>
<td></td>
<td>- Medium</td>
</tr>
<tr>
<td></td>
<td>- Low</td>
</tr>
<tr>
<td></td>
<td>- Informational</td>
</tr>
<tr>
<td>operating_systems</td>
<td>An object detailing the operating systems to which the Detection Rule can be applied. It will include the &quot;name&quot; field. This can consist of:</td>
</tr>
<tr>
<td></td>
<td>- &quot;Windows&quot;</td>
</tr>
<tr>
<td></td>
<td>- &quot;MacOS&quot;</td>
</tr>
<tr>
<td>date_added</td>
<td>The timestamp (in UTC) when the Detection Rule was added to the tenant.</td>
</tr>
<tr>
<td>enabled</td>
<td>Determines whether or not a Detection Rule is enabled in the Detection Rule Set. When viewing the content of a Detection Rule Set, this should always be set to 'true'.</td>
</tr>
<tr>
<td>notification_enabled</td>
<td>Determines whether or not the message defined in the 'notification_message' field should display on the device when the Detection Rule is triggered.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> To enable Display Desktop Notification on Device using the API, set notification_enabled and DisplayDesktopNotification to &quot;true&quot;. To disable, set both to &quot;false&quot;. The DisplayDesktopNotification setting enables or disables the feature. The notification_enabled setting affects the Display Desktop Notification on Device checkbox in the Console as enabled (checked) or disabled (unchecked).</td>
</tr>
<tr>
<td>responses</td>
<td>A list of response objects for each Response Action enabled for a particular Detection Rule. Each object will include the following fields:</td>
</tr>
</tbody>
</table>
### Field Name | Description
--- | ---
| | ■ template_id: The ID of the response template to use (this is provided by Cylance).
| | ■ response_rule_id: The ID of the response rule to enable (these are provided by Cylance).
| | ■ response_rule_version: The version of the response rule to enable (this is provided by Cylance).
| | ■ description: The description / name of the response rule.
| | ■ value: A currently unused field.
| | ■ enabled: This will always be 'true' when viewing a Detection Rule Set.
| | ■ created: The date that the Response Rule was added to the tenant.

**exceptions**  
A list of Exception Rule objects that should be applied to the Detection Rule. Each object will include the following fields:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>
| | ■ exception_id: The unique ID of the Exception Rule.
| | ■ enabled: This will always be 'true' when viewing a Detection Rule Set.
| | ■ name: The name of the Exception Rule.

**playbooks**  
A list of Package Playbook unique IDs that will be executed when the Detection Rule is triggered on the device.

---

**Delete Detection Rule Set**

Delete a Detection Rule Set.

**Service Endpoint:**

- `/rulesets/v2/{ruleset_id}`

For example: `https://protectapi.cylance.com/rulesets/v2/c407f28a-3805-4014-b32c-0c2553ac1e17`

**Method:**

- HTTP/1.1 DELETE

**Request Headers:**

- Accept: application/json
- Authorization: Bearer `<JWT Token returned by Auth API>` with the `opticsruleset:delete` scope encoded

**Request**

None
Delete Multiple Detection Rule Sets

Delete multiple Detection Rule Sets in a single request.

Service Endpoint:

- /rulesets/v2

For example: https://protectapi.cylance.com/rulesets/v2

Method:

- HTTP/1.1 DELETE

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsruleset:delete scope encoded

Request

```
{
  "ids": [
    "c407f28a-3805-4014-b32c-0c2553ac1e17",
    "998ece50-49af-472a-b0d8-3c3700883736"
  ]
}
```

Response

200 OK - Rulesets deleted

```
[
  {
    "id": "",
```
"success": true,
"message": ""
}
]

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resources page requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

Request JSON Schema Description

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ids</td>
<td>A list of Detection Rule Set IDs to be deleted.</td>
</tr>
</tbody>
</table>

Response JSON Schema Description

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>A Detection Rule Set ID that was attempted to be deleted.</td>
</tr>
<tr>
<td>success</td>
<td>A Boolean field denoting whether or not the Detection Rule Set was deleted.</td>
</tr>
<tr>
<td>message</td>
<td>A string containing any error, success, or warning messages.</td>
</tr>
</tbody>
</table>
Detection Exceptions
The CylanceOPTICS Detection Exceptions API allows users to add exceptions to their detection rules. Users can create a Detection Exception from a false positive detection, from the Detection Summary page, and from the Detection Details page.

The CylanceOPTICS Detection Exceptions API includes:

- Getting the content for a Detection Exception
- Getting a list of Detection Exceptions for a tenant
- Getting a list of Detection Exceptions as a CSV file
- Creating a Detection Exception
- Updating a Detection Exception
- Deactivating (or soft deleting) a Detection Exception

### Get Detection Exceptions List

Retrieve a list of Detection Exception rules available in a tenant.

**Service Endpoint:**

- `/exceptions/v2`

For example: https://protectapi.cylance.com/exceptions/v2

**Method:**

- HTTP/1.1 GET

**Request Headers:**

- `Accept: application/json`
- `Authorization: Bearer <JWT Token returned by Auth API> with the opticsexception:list scope encoded`

**Request**

None

**Response**

200 OK

```json
{
  "page_size": 1,
  "total_pages": 1,
  "page_items": [
    {
      "Id": "99eff732-4d39-47df-b246-f7dbb8a8fd86",
    }
  ]
}
```
"Name": "AO Exception",
"Version": 1,
"RulesetCount": 1,
"LastModified": "2019-04-02T17:58:03Z",
"PolicyCount": 1,
"DeviceCount": 1,
"OperatingSystem": [
  {
    "Name": "macOS"
  }
],
"ModifiedBy": {
  "id": "a2c0ac7a-a63d-4583-b646-ae10db9c9769",
  "login": "testuser@email.com"
},
"Description": "Instigating process"
},
"total_number_of_items": 1,
"page_number": 1
}

400 Bad Request - Malformed request.
401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
404 Not Found - No such resource found.
500 Internal Server Error - An unforeseeable error has occurred.
503 Service Unavailable - Unable to respond at this time, please try again later.

Response JSON Schema Description

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>page_size</td>
<td>The number of items on the page.</td>
</tr>
<tr>
<td>total_pages</td>
<td>The total number of pages of this size.</td>
</tr>
<tr>
<td>total_number_of_items</td>
<td>The total number of Exceptions in the tenant.</td>
</tr>
<tr>
<td>page_number</td>
<td>The current page number of results.</td>
</tr>
<tr>
<td>page_items</td>
<td>A list of Exception objects that are available in the tenant.</td>
</tr>
<tr>
<td>id</td>
<td>The unique ID of the Exception.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the Exception.</td>
</tr>
<tr>
<td>Description</td>
<td>The description of the Exception.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DeviceCount</td>
<td>The number of devices that have the Exception applied.</td>
</tr>
<tr>
<td>LastModified</td>
<td>The timestamp (in UTC) of the last time that the Exception was modified.</td>
</tr>
<tr>
<td>ModifiedBy</td>
<td>The last user to modify the Exception.</td>
</tr>
<tr>
<td>OperatingSystem</td>
<td>The operating systems that the Exception can be applied to. The &quot;name&quot; field can consist of:</td>
</tr>
<tr>
<td></td>
<td>- &quot;Windows&quot;</td>
</tr>
<tr>
<td></td>
<td>- &quot;MacOS&quot;</td>
</tr>
<tr>
<td>PolicyCount</td>
<td>The number of policies that have the Exception applied.</td>
</tr>
<tr>
<td>RulesetCount</td>
<td>The number of Detection Rule Sets that have the Exception applied.</td>
</tr>
<tr>
<td>Version</td>
<td>The version of the Exception.</td>
</tr>
</tbody>
</table>

**Get Detection Exception CSV List**

Retrieve a CSV where every line represents an Exception rule available in the tenant.

**Service Endpoint:**

- /rulesets/v2/csv

For example: [https://protectapi.cylance.com/rulesets/v2/csv](https://protectapi.cylance.com/rulesets/v2/csv)

**Method:**

- HTTP/1.1 GET

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsexception:list scope encoded

**Request**

None

**Response**

200 OK - A proper response displays the requested detection information in CSV format.

400 Bad Request - Malformed request.
401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
404 Not Found - No such resource found.
500 Internal Server Error - An unforeseeable error has occurred.
503 Service Unavailable - Unable to respond at this time, please try again later.

Response JSON Schema Description

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the Exception.</td>
</tr>
<tr>
<td>Id</td>
<td>The unique ID of the Exception.</td>
</tr>
<tr>
<td>Version</td>
<td>The version of the Exception.</td>
</tr>
<tr>
<td>Description</td>
<td>The description of the Exception.</td>
</tr>
<tr>
<td>Last Modified</td>
<td>The timestamp (in UTC) of the last time the Exception was modified.</td>
</tr>
<tr>
<td>Modified By</td>
<td>The email address of the user who last modified the Exception.</td>
</tr>
<tr>
<td>Device Count</td>
<td>The number of devices that have the Exception applied.</td>
</tr>
<tr>
<td>Ruleset Count</td>
<td>The number of Detection Rule Sets that have the Exception enabled.</td>
</tr>
</tbody>
</table>

Get Detection Exception Content

Retrieve the content of an Exception in its native JSON structure. For more information about the structure of CylanceOPTICS Exception Rules, refer to the "Custom Rule Guide" on page 331.

Service Endpoint:

- /exceptions/v2/{exception_id}

For example: https://protectapi.cylance.com/exceptions/v2/24eff732-4d39-47df-b246-f7dbb8a8fd87

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsexception:read scope encoded
Request
None
Response
200 OK

```json
{
   "ObjectType": "ExceptionRule",
   "Plugin": {
      "Name": "OpticsDetector"
   },
   "Tags": [
      "CylanceOPTICS",
      "Exception"
   ],
   "OperatingSystems": [
      {
         "Name": "Windows"
      }
   ],
   "SchemaVersion": 1,
   "Product": {
      "Name": "CylanceOPTICS"
   },
   "States": [
      {
         "Name": "Exception",
         "Scope": "Global",
         "FieldOperators": {
            "0": {
               "Type": "Equals",
               "OperandType": "String",
               "Options": {
                  "IgnoreCase": true
               },
               "Operands": [
                  {
                     "Source": "InstigatingProcess",
                     "Data": "Name"
                  },
                  {
                     "Source": "LiteralSet",
                     "Data": [
                        "old.exe",
                        "new.exe"
                     ]
                  }
               ]
            }
         },
         "Function": "0"
      }
   }
}
```
Create Detection Exception

Create a new Detection Exception by sending the native JSON structure of a Detection Exception. For more information about the structure of CylanceOPTICS Exception Rules, refer
to "Custom Rule Guide" on page 331.

**Service Endpoint:**

- /exceptions/v2

For example: [https://protectapi.cylance.com/exceptions/v2](https://protectapi.cylance.com/exceptions/v2)

**Method:**

- HTTP/1.1 POST

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsexception:create scope encoded

**Request**

```json
{
    "Name": "My Exception",
    "Description": "My Exception Description",
    "ObjectType": "ExceptionRule",
    "OperatingSystems": [
        { "Name": "Windows" }
    ],
    "Plugin": {
        "Name": "OpticsDetector"
    },
    "Product": {
        "Name": "CylanceOPTICS"
    },
    "SchemaVersion": 1,
    "States": [
        { "Name": "UnsignedProc",
          "Scope": "Global",
          "Function": "Function",
          "FieldOperators": {
              "Function": { "Type": "EqualsAny",
                            "Operands": [ { "Source": "LiteralSet",
                                            "Data": iexplore.exe" } ],
                            "OperandType": "string",
                            "Options": { ... }}... ```
"IgnoreCase": true

},

"Actions": [
{
"Type": "AOI",
"ItemName": "InstigatingProcess",
"Position": "PostActivation"
}
]
],

"Tags": [
"CylanceOPTICS, Exception"
]
}

Response

200 OK - ExceptionRule created successfully

{
"Id": "24eff732-4d39-47df-b246-f7dbb8a8fd87",
"Name": "My Exception",
"Description": "My Exception Description",
"ObjectType": "ExceptionRule",
"OperatingSystems": [
{
"Name": "Windows"
}
],
"Plugin": {
"Name": "OpticsDetector"
},
"Product": {
"Name": "CylanceOPTICS"
},
"SchemaVersion": 1,
"States": [
{
"Name": "UnsignedProc",
"Scope": "Global",
"Function": "Function",
"FieldOperators": {
"Function": {
"Type": "EqualsAny",
"Operands": [
{
"Source": "LiteralSet",
"Data": "iexplore.exe"
}]
}]
]
400 Bad Request - Malformed request.
401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
404 Not Found - No such exception found.
500 Internal Server Error - An unforeseeable error has occurred.
503 Service Unavailable - Unable to respond at this time, please try again later.

**Note:** The "id" and "version" fields are automatically populated when the request is submitted.

**Request and Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The description for the Detection Exception.</td>
</tr>
<tr>
<td>Id</td>
<td>The unique identifier for the Detection Exception.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the Detection Exception.</td>
</tr>
<tr>
<td>ObjectType</td>
<td>The type of object defined in this rule.</td>
</tr>
<tr>
<td></td>
<td>- DetectionRule</td>
</tr>
<tr>
<td></td>
<td>- ResponseRule</td>
</tr>
<tr>
<td>OperatingSystems</td>
<td>The list of operating systems to which the Detection Exception applies.</td>
</tr>
</tbody>
</table>
### Update Detection Exception

Update a Detection Exception by sending a new JSON structure. For more information about the structure of CylanceOPTICS Exception Rules, refer to "Custom Rule Guide" on page 331.

**Service Endpoint:**

- `/exceptions/v2/{exception_id}`

For example: https://protectapi.cylance.com/exceptions/v2/24eff732-4d39-47df-b246-f7dbb8a8fd87

**Method:**

- HTTP/1.1 PUT

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsexception:update scope encoded

**Request**

```json
{
   "Name": "My Exception",
   "Description": "My Exception Description",
   "ObjectType": "ExceptionRule",
   "OperatingSystems": [
      {
         "Name": "Windows"
      }
   ]
}
```
"Plugin": {
  "Name": "OpticsDetector"
},
"Product": {
  "Name": "CylanceOPTICS"
},
"SchemaVersion": 1,
"States": [
  {
    "Name": "UnsignedProc",
    "Scope": "Global",
    "Function": "Function",
    "FieldOperators": {
      "Function": {
        "Type": "EqualsAny",
        "Operands": [
          {
            "Source": "LiteralSet",
            "Data": "iexplore.exe"
          }
        ],
        "OperandType": "string",
        "Options": {
          "IgnoreCase": true
        }
      }
    },
    "OperandType": "string",
    "Options": {
      "IgnoreCase": true
    }
  },
  "Actions": [
    {
      "Type": "AOI",
      "ItemName": "InstigatingProcess",
      "Position": "PostActivation"
    }
  ],
  "Tags": [
    "CylanceOPTICS, Exception"
  ]
}]

Response

202 OK - ExceptionRule updated successfully

{
  "Id": "24eff732-4d39-47df-b246-f7dbb8a8fd87",
  "Name": "My Exception",
  "Description": "My Exception Description",
  "ObjectType": "ExceptionRule",
  "OperatingSystems": [
    {
      "Name": "Windows",
      "Version": "10.0.18363"
    }
  ]
}
"Name": "Windows"
}
],
"Plugin": {
  "Name": "OpticsDetector"
},
"Product": {
  "Name": "CylanceOPTICS"
},
"States": [],
"SchemaVersion": 1,
"States": [
  {
    "Name": "UnsignedProc",
    "Scope": "Global",
    "Function": "Function",
    "FieldOperators": {
      "Function": {
        "Type": "EqualsAny",
        "Operands": [
          {
            "Source": "LiteralSet",
            "Data": "iexplore.exe"
          }
        ],
        "OperandType": "string",
        "Options": {
          "IgnoreCase": true
        }
      }
    },
    "OperandType": "string",
    "Options": {
      "IgnoreCase": true
    }
  },
  "Actions": [
    {
      "Type": "AOI",
      "ItemName": "InstigatingProcess",
      "Position": "PostActivation"
    }
  ],
  "Tags": [
    "CylanceOPTICS, Exception"
  ],
  "Version": 2
}

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such exception found.
500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

**Note:** The "id" and "version" fields are automatically populated when the request is submitted.

### Request and Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The description for the Detection Exception.</td>
</tr>
<tr>
<td>Id</td>
<td>The unique identifier for the Detection Exception.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Part of the Response, after the Detection Exception is updated.</td>
</tr>
<tr>
<td>Name</td>
<td>The name of the Detection Exception.</td>
</tr>
<tr>
<td>ObjectType</td>
<td>The type of object defined in this rule.</td>
</tr>
<tr>
<td></td>
<td>- DetectionRule</td>
</tr>
<tr>
<td></td>
<td>- ResponseRule</td>
</tr>
<tr>
<td>OperatingSystems</td>
<td>The list of operating systems to which the Detection Exception applies.</td>
</tr>
<tr>
<td>Plugin</td>
<td>The name of the product feature to which the Detection Exception applies.</td>
</tr>
<tr>
<td>Product</td>
<td>The name of the Cylance product to which the Detection Exception applies.</td>
</tr>
<tr>
<td>SchemaVersion</td>
<td>The version of the schema.</td>
</tr>
<tr>
<td>States</td>
<td>The list of all available states. If no paths are specified, the states are</td>
</tr>
<tr>
<td></td>
<td>transitioned in the order they are specified.</td>
</tr>
<tr>
<td></td>
<td>**See the <a href="#">&quot;Custom Rule Guide&quot; on page 331</a> for information about States.</td>
</tr>
<tr>
<td>Tags</td>
<td>The list of tags associated with the Detection Exception.</td>
</tr>
<tr>
<td>Version</td>
<td>The version number for the Detection Exception.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Part of the Response, after the Detection Exception is updated.</td>
</tr>
</tbody>
</table>

### Deactivate / Delete Detection Exception

Deactivate (or "soft delete") a Detection Exception and remove it from the Detection Rule Sets list.

**Service Endpoint:**

- `/exceptions/v2/{exception_id}/deactivate`

For example: https://protectapi.cylance.com/exceptions/v2/24eff732-4d39-47df-b246-f7dbb8a8fd87/deactivate

**Method:**
HTTP/1.1 POST

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsexception:update scope encoded

Request

None

Response

200 OK

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such exception found.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

Note: Detection Rule Sets are not automatically communicated to all endpoints when updates to Detection Exceptions are made. To ensure that the latest logic is applied to endpoints in the quickest manner, re-save any affected Detection Rule Sets (either via the UI or API).
Device Commands
Device Commands

The CylanceOPTICS Device Commands API allows users to perform actions on the endpoint. For example, locking down an endpoint or retrieving a file from an endpoint.

The CylanceOPTICS Device Commands API includes:

- Locking down an endpoint
- Getting Device Lockdown History for a tenant
- Requesting a file retrieval from an endpoint
- Checking the file retrieval status for an endpoint
- Getting the retrieved file results

Lockdown Device Command

Create a CylanceOPTICS Device Lockdown command resource for a specific device.

Service Endpoint:

```
/devicecommands/v2/{{device_id}}/lockdown?value=true&expires=d:hh:mm
```

For example:

```
https://protectapi.cylance.com/devicecommands/v2/45E07F34E76B4A9EB167D6D0C510D6BA/lockdown?value=true&expires=0:00:05
```

Method:

- HTTP/1.1 PUT

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticscommand:create scope encoded

Request

Append the following optional query string parameters:

- value: Whether to lockdown or not. Default is 'true'.
- expires: Duration of the lockdown. Format is 'd:hh:mm'.
  - Maximum is 3 days. Example: expires=3
  - Minimum is 5 minutes. Example: expires=0:00:05

Response
201 OK - Command created

```
{
    "id": "45E07F34E76B4A9EB167D6D0C510D6BA",
    "hostname": "User-Laptop-A123",
    "tenant_id": "4b1640d2-d563-41cf-94a7-0dalda6aa98",
    "connection_status": "connected",
    "optics_device_version": "2.3.2050.930",
    "password": "foo",
    "lockdown_expiration": "2019-01-01T00:00:05Z",
    "lockdown_initiated": "2019-01-01T00:00:00Z",
    "lockdown_history": [
        {
            "user_id": "a2c0ac7a-a63d-4583-b646-a10db9768",
            "timestamp": "2019-01-01T00:00:00Z",
            "command": "ClearLockdown"
        }
    ]
}
```

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such exception found.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The unique device ID that the lockdown command was issued to.</td>
</tr>
<tr>
<td>hostname</td>
<td>The hostname of the device that the lockdown command was issued to.</td>
</tr>
<tr>
<td>tenant_id</td>
<td>The unique tenant ID of the tenant that the device belongs to.</td>
</tr>
<tr>
<td>connection_status</td>
<td>Displays whether or not the device is connected to Cylance's cloud services.</td>
</tr>
<tr>
<td>optics_device_version</td>
<td>Returns the numerical version of CylanceOPTICS running on the device.</td>
</tr>
<tr>
<td>password</td>
<td>The password required to unlock the device.</td>
</tr>
<tr>
<td>lockdown_expiration</td>
<td>The timestamp (in UTC) of when the current device lockdown is set to expire.</td>
</tr>
<tr>
<td>lockdown_initiated</td>
<td>The timestamp (in UTC) of when the current device lockdown was initiated.</td>
</tr>
<tr>
<td>lockdown_history</td>
<td>A list of historical device lockdown commands issued to this particular device.</td>
</tr>
</tbody>
</table>
### Field Name | Description
--- | ---
user_id | The unique ID of the user who locked down the device.
timestamp | The timestamp (in UTC) of when the command was initiated.
command | The command that was executed.

### Get Device Lockdown History

Request the current lockdown state and lockdown history for a specific device.

**Service Endpoint:**

- `/devicecommands/v2/{{device_id}}/lockdown`

For example:

https://protectapi.cylance.com/devicecommands/v2/45E07F34E76B4A9EB167D6D0C510D6BA/lockdown

**Method:**

- HTTP/1.1 GET

**Request Headers:**

- `Accept: application/json`
- `Authorization: Bearer <JWT Token returned by Auth API> with the opticscommand:read scope encoded`

**Request**

None

**Response**

200 OK

```
{
    "id": "45E07F34E76B4A9EB167D6D0C510D6BA",
    "hostname": "User-Laptop-A123",
    "tenant_id": "4b1640d2-d563-41cf-94a7-0da1dca6aa98",
    "connection_status": "connected",
    "optics_device_version": "2.3.2050.930",
    "password": "foo",
    "lockdown_expiration": "2019-01-01T00:00:05Z",
    "lockdown_initiated": "2019-01-01T00:00:00Z",
    "lockdown_history": [
        {
            "user_id": "a2c0ac7a-a63d-4583-b646-ae10db9c9768",
```
400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such exception found.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The unique device ID that the lockdown command was issued to.</td>
</tr>
<tr>
<td>hostname</td>
<td>The hostname of the device that the lockdown command was issued to.</td>
</tr>
<tr>
<td>tenant_id</td>
<td>The unique tenant ID of the tenant that the device belongs to.</td>
</tr>
<tr>
<td>connection_status</td>
<td>Displays whether or not the device is connected to Cylance’s cloud services.</td>
</tr>
<tr>
<td>optics_device_version</td>
<td>Returns the numerical version of CylanceOPTICS running on the device.</td>
</tr>
<tr>
<td>password</td>
<td>The password required to unlock the device.</td>
</tr>
<tr>
<td>lockdown_expiration</td>
<td>The timestamp (in UTC) of when the current device lockdown is set to expire.</td>
</tr>
<tr>
<td>lockdown_initiated</td>
<td>The timestamp (in UTC) of when the current device lockdown was initiated.</td>
</tr>
<tr>
<td>lockdown_history</td>
<td>A list of historical device lockdown commands issued to this particular device.</td>
</tr>
<tr>
<td>user_id</td>
<td>The unique ID of the user who locked down the device.</td>
</tr>
<tr>
<td>timestamp</td>
<td>The timestamp (in UTC) of when the command was initiated.</td>
</tr>
<tr>
<td>command</td>
<td>The command that was executed.</td>
</tr>
</tbody>
</table>

**Get Retrieved File Results**

Obtain a history of file retrieval requests for all devices in the tenant.

**Service Endpoint:**
For example: https://protectapi.cylance.com/devicecommands/v2/retrieved_files?page=1&page_size=100

Method:
- HTTP/1.1 GET

Request Headers:
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticscommand:read scope encoded

Request

Append the following query string parameters:
- q: Case-insensitive search term.
- page: The page number to request. Defaults to 1.
- page_size: The number of file retrieval records to retrieve per page. Defaults to 20.
- sort: Sort by field (adding '-' in front of the value denotes descending order).

Response

200 OK

```json
[
  {
    "tenant_id": "4b1640d2-d563-41cf-94a7-0daldac6aa98",
    "user_id": "a2c0ac7a-a63d-4583-b646-ae10db9c9768",
    "device_id": "45E07F34E76B4A9EB167D6D0C510D6BA",
    "created_at": "2019-01-01T00:00:00Z",
    "filepath": "C:\path\to\file.txt",
    "download_url": "https://unique/url",
    "file_status": "PENDING",
    "file_status_description": "Too Large | Does Not Exist",
    "password": "foo",
    "md5": "d41d8cd98f00b204e9800998ecf8427e",
    "sha1": "da39a3ee5e6b4b0d3255bfe9f95601890af8079",
    "sha256": "bf17366ee3bb8068a9ad70fc9e68496e7e311a055bf4ffeef53cc5d29ccce52",
    "correlation_id": "00000000000000000000000000000000",
    "user_login": "testuser@email.com",
    "hostname": "User-Laptop-A123"
  }
]
```
400 Bad Request - Returned for the following reasons:

- The tenant ID could not be retrieved from the JWT token specified in the Authorization header.
- The unique identifier for the execution is not valid.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such exception found.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>An object containing the various fields associated with the file retrieval request.</td>
</tr>
<tr>
<td>tenant_id</td>
<td>The unique tenant ID of the tenant that the device belongs to.</td>
</tr>
<tr>
<td>user_id</td>
<td>The unique ID of the user who locked down the device.</td>
</tr>
<tr>
<td>device_id</td>
<td>The unique device ID that the lockdown command was issued to.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> See &quot;About Device ID on page 25&quot; for device ID formatting.</td>
</tr>
<tr>
<td>created_at</td>
<td>The timestamp (in UTC) of when the file retrieval was requested.</td>
</tr>
<tr>
<td>filepath</td>
<td>The file path of the requested file.</td>
</tr>
<tr>
<td>download_url</td>
<td>The unique URL and parameters required to download the retrieved file.</td>
</tr>
<tr>
<td>file_status</td>
<td>The status of the file retrieval. This will always be &quot;PENDING&quot; for newly created file retrievals.</td>
</tr>
<tr>
<td></td>
<td>- REQUEST: The file retrieval has not been requested, but the user may issue a request for it.</td>
</tr>
<tr>
<td></td>
<td>- RETRY_REQUEST: The file retrieval has been requested previously but no results were received. It can be requested again.</td>
</tr>
<tr>
<td></td>
<td>- PENDING: The file retrieval has been requested but has not yet been completed.</td>
</tr>
<tr>
<td></td>
<td>- DOES_NOT_EXIST: The file retrieval has been requested but is not present on the device.</td>
</tr>
<tr>
<td></td>
<td>- AVAILABLE: The file is available for download. A download link (download_url) is generated and valid for the next 10 minutes.</td>
</tr>
<tr>
<td></td>
<td>- UNAVAILABLE: The file is not available. This status may indicate that the requested device is not online, or the requested device failed to upload the file. This status will become RETRY_REQUEST after an hour.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>file_status_description</td>
<td>Displays any errors or status messages associated with the retrieval request.</td>
</tr>
<tr>
<td>password</td>
<td>The password required to decrypt the retrieved file.</td>
</tr>
<tr>
<td>md5</td>
<td>The MD5 hash of the retrieved file.</td>
</tr>
<tr>
<td>sha1</td>
<td>The SHA1 hash of the retrieved file.</td>
</tr>
<tr>
<td>sha256</td>
<td>The SHA256 hash of the retrieved file.</td>
</tr>
<tr>
<td>correlation_id</td>
<td>The correlation ID associated with this action.</td>
</tr>
<tr>
<td>user_login</td>
<td>The email address of the user who initiated the file retrieval request.</td>
</tr>
<tr>
<td>hostname</td>
<td>The hostname of the device that the file retrieval was requested on.</td>
</tr>
</tbody>
</table>

**Request File Retrieval From Device**

Request that the specified file be retrieved from a specified device and stored in the Cylance Console for later analysis.

**Service Endpoint:**

```
/devicecommands/v2/{{device_id}}/getfile
```

For example:

https://protectapi.cylance.com/devicecommands/v2/45E07F34E76B4A9EB167D6D0C510D6BA/getfile

**Method:**

```
HTTP/1.1 POST
```

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticscommand:read
  scope encoded

**Request**

```json
{
  "file_path": "C:\path\to\file.txt"
}
```

**Response**
400 Bad Request - Returned for the following reasons:

- The tenant ID could not be retrieved from the JWT token specified in the Authorization header.
- The page number or page size specified are less than or equal to zero.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such exception found.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>An object containing the various fields associated with the file retrieval request.</td>
</tr>
<tr>
<td>tenant_id</td>
<td>The unique tenant ID of the tenant that the device belongs to.</td>
</tr>
<tr>
<td>user_id</td>
<td>The unique ID of the user who locked down the device.</td>
</tr>
<tr>
<td>device_id</td>
<td>The unique device ID that the lockdown command was issued to. <strong>Note:</strong> See &quot;About Device ID on page 25&quot; for device ID formatting.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>created_at</td>
<td>The timestamp (in UTC) of when the file retrieval was requested.</td>
</tr>
<tr>
<td>filepath</td>
<td>The file path of the requested file.</td>
</tr>
<tr>
<td>download_url</td>
<td>The unique URL and parameters required to download the retrieved file.</td>
</tr>
<tr>
<td>file_status</td>
<td>The status of the file retrieval. This will always be &quot;PENDING&quot; for newly created file retrievals.</td>
</tr>
<tr>
<td>file_status_description</td>
<td>Displays any errors or status messages associated with the retrieval request.</td>
</tr>
<tr>
<td>password</td>
<td>The password required to decrypt the retrieved file.</td>
</tr>
<tr>
<td>md5</td>
<td>The MD5 hash of the retrieved file.</td>
</tr>
<tr>
<td>sha1</td>
<td>The SHA1 hash of the retrieved file.</td>
</tr>
<tr>
<td>sha256</td>
<td>The SHA256 hash of the retrieved file.</td>
</tr>
<tr>
<td>correlation_id</td>
<td>The correlation ID associated with this action.</td>
</tr>
<tr>
<td>user_login</td>
<td>The email address of the user who initiated the file retrieval request.</td>
</tr>
<tr>
<td>hostname</td>
<td>The hostname of the device that the file retrieval was requested on.</td>
</tr>
</tbody>
</table>

### Check File Retrieval Status From Device

Check the status of a previously requested file retrieval operation.

**Service Endpoint:**

- /devicecommands/v2/{{device_id}}/getfile:get

For example:

https://protectapi.cylance.com/devicecommands/v2/45E07F34E76B4A9EB167D6D0C510D6BA/getfile:get

**Method:**

- HTTP/1.1 POST

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticscommand:read scope encoded

**Request**
Response

200 OK

{  "data": {    "tenant_id": "4b1640d2-d563-41cf-94a7-0da1dca6a98",    "user_id": "a2c0ac7a-a63d-4583-b646-ae10db9c9768",    "device_id": "45E07F34E76B4A9EB167D6D0C510D6BA",    "created_at": "2019-01-01T00:00:00Z",    "filepath": "C:\path\to\file.txt",    "download_url": "https://unique/url",    "file_status": "PENDING",    "file_status_description": "Too Large | Does Not Exist",    "password": "foo",    "md5": "d41d8cd98f00b204e9800998ecf8427e",    "sha1": "da39a3ee5e6b4b0d3255bfef95601890af80709",    "sha256": "bf17366ee3bb8068a9ad70fc9e68496e7e311a055bf4ffeef53cc5d29ccce52",    "correlation_id": "00000000000000000000000000000000",    "user_login": "testuser@email.com",    "hostname": "User-Laptop-A123"  }
}

400 Bad Request - Returned for the following reasons:

- The tenant ID could not be retrieved from the JWT token specified in the Authorization header.
- The page number or page size specified are less than or equal to zero.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such exception found.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

Response JSON Schema Descriptions
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>An object containing the various fields associated with the file retrieval request.</td>
</tr>
<tr>
<td>tenant_id</td>
<td>The unique tenant ID of the tenant that the device belongs to.</td>
</tr>
<tr>
<td>user_id</td>
<td>The unique ID of the user who locked down the device.</td>
</tr>
<tr>
<td>device_id</td>
<td>The unique device ID that the lockdown command was issued to. <strong>Note</strong>: See &quot;About Device ID&quot; on page 25 for device ID formatting.</td>
</tr>
<tr>
<td>created_at</td>
<td>The timestamp (in UTC) of when the file retrieval was requested.</td>
</tr>
<tr>
<td>filepath</td>
<td>The file path of the requested file.</td>
</tr>
<tr>
<td>download_url</td>
<td>The unique URL and parameters required to download the retrieved file.</td>
</tr>
<tr>
<td>file_status</td>
<td>The status of the file retrieval. This will always be &quot;PENDING&quot; for newly created file retrievals.</td>
</tr>
<tr>
<td>file_status_description</td>
<td>Displays any errors or status messages associated with the retrieval request.</td>
</tr>
<tr>
<td>password</td>
<td>The password required to decrypt the retrieved file.</td>
</tr>
<tr>
<td>md5</td>
<td>The MD5 hash of the retrieved file.</td>
</tr>
<tr>
<td>sha1</td>
<td>The SHA1 hash of the retrieved file.</td>
</tr>
<tr>
<td>sha256</td>
<td>The SHA256 hash of the retrieved file.</td>
</tr>
<tr>
<td>correlation_id</td>
<td>The correlation ID associated with this action.</td>
</tr>
<tr>
<td>user_login</td>
<td>The email address of the user who initiated the file retrieval request.</td>
</tr>
<tr>
<td>hostname</td>
<td>The hostname of the device that the file retrieval was requested on.</td>
</tr>
</tbody>
</table>
Focus View
Focus View

The CylanceOPTICS Focus View API allows users to retrieve an information trail starting with the first event related to an artifact from an InstaQuery result or a CylancePROTECT event.

The CylanceOPTICS Focus View API includes:

- Searching for Focus View results
- Generating a Focus View
- Getting a summary of a Focus View
- Getting the results of a Focus View
- Getting a list of Focus Views that have been made in a tenant

Get Focus View List

Retrieve a list of Focus Views that have been made in the tenant.

Service Endpoint:

- /foci/v2?page=m&page_size=n

For example: https://protectapi.cylance.com/foci/v2?page=1&page_size=100

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsfocus:read scope encoded

Request

Note: The 'q' request parameter was replaced with multiple request parameters to provide more flexibility when filtering the Focus View List. Any Get Focus View List requests that contain the 'q' request parameter will not return any results. Requests should use the following parameters.

Append the following query string parameters:

- artifact_type: The type of Artifact for the Focus View. Types include Protect, Process, File, and NetworkConnection. The artifact type is case-insensitive.
- **created_at**: The date when the file retrieval was requested. The date format is YYYY-MM-DD. The results are for a 24 hour period. For example, using "&created_at=2019-11-01" will return results that occurred from 2019-11-01:00:00:00 to 2019-11-01:23:59:59.

- **description**: The human-readable description for the Focus View. The description is case-insensitive.

- **hostname**: The hostname of the device for which the retrieval was requested. The hostname is case-insensitive.

- **status**: The status of the Focus View request or result. Statuses include AVAILABLE, DOES_NOT_EXIST, PENDING, REQUEST, RETRY_REQUEST, UNAVAILABLE, and UNKNOWN DEVICE. The statuses are case-sensitive.

- **page**: The page number to request. Defaults to 1.

- **page_size**: The number of file retrieval records to retrieve per page. Defaults to 20.

- **sort**: Sort by field (adding '-' in front of the value denotes descending order).

**Response**

200 OK

```json
{
    "page_size": 10,
    "total_pages": 5,
    "page_items": [
    
    {
        "device_id": "E378DACB9324453AB8C65A8406952195",
        "artifact_type": "Process",
        "artifact_subtype": "Uid",
        "value": "59F849F29BEF1F889AAF50F9153618",
        "threat_type": "THREAT",
        "descriptions": "Focus View Example",
        "id": "A0AC3D2117C40D0576C6D099069E96G",
        "tenant_id": "4B1640D2D56341CF94A70DA1DCA6AA98",
        "created_at": "2018-07-26T01:20:07.596Z",
        "hostname": "User-Laptop-A123",
        "status": "AVAILABLE",
        "relations": [
        
        {
            "object": "/survey/survey_id",
            "relationship": "origin-of"
        }
        
        
        ]
    
    }
    
    "total_number_of_items": 47,
    "page_number": 1
}
```
400 Bad Request - Malformed request.
401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
404 Not Found - No such exception found.
500 Internal Server Error - An unforeseeable error has occurred.
503 Service Unavailable - Unable to respond at this time, please try again later.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>page_size</td>
<td>The number of items per page.</td>
</tr>
<tr>
<td>total_pages</td>
<td>The total number of pages of this page size.</td>
</tr>
<tr>
<td>total_number_of_items</td>
<td>The total number of Focus Views available in the tenant.</td>
</tr>
<tr>
<td>page_number</td>
<td>The current page number.</td>
</tr>
<tr>
<td>page_items</td>
<td>A list of Focus View objects.</td>
</tr>
<tr>
<td>device_id</td>
<td>The unique device ID that the lockdown command was issued to.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> See &quot;About Device ID on page 25&quot; for device ID formatting.</td>
</tr>
<tr>
<td>artifact_type</td>
<td>The type of Artifact for the Focus View.</td>
</tr>
<tr>
<td></td>
<td>▪ Protect: Request a Focus View for a CylancePROTECT-generated event.</td>
</tr>
<tr>
<td></td>
<td>▪ Process: Request a Focus View for a Process artifact to visualize how a</td>
</tr>
<tr>
<td></td>
<td>process interacts with the device. This is the most common option.</td>
</tr>
<tr>
<td></td>
<td>▪ File: Request a Focus View for a File artifact to visualize how the file</td>
</tr>
<tr>
<td></td>
<td>has been interacted with.</td>
</tr>
<tr>
<td></td>
<td>▪ NetworkConnection: Request a Focus View for a Network artifact to visualize</td>
</tr>
<tr>
<td></td>
<td>communications associated with an IP address.</td>
</tr>
<tr>
<td>artifact_subtype</td>
<td>This field should always be &quot;Uid&quot; at this time.</td>
</tr>
<tr>
<td>value</td>
<td>The UID of the Artifact used to gather the Focus View.</td>
</tr>
<tr>
<td>threat_type</td>
<td>An option field to use with a &quot;Protect&quot; artifact_type to denote the type of</td>
</tr>
<tr>
<td></td>
<td>threat that a Focus View is being generated for.</td>
</tr>
<tr>
<td>description</td>
<td>The human-readable description for the Focus View.</td>
</tr>
<tr>
<td>id</td>
<td>The unique ID of the Focus View.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>tenant_id</td>
<td>The unique tenant ID of the tenant that the device belongs to.</td>
</tr>
<tr>
<td>created_at</td>
<td>The timestamp (in UTC) of when the file retrieval was requested.</td>
</tr>
<tr>
<td>hostname</td>
<td>The hostname of the device that the file retrieval was requested on.</td>
</tr>
</tbody>
</table>
| status     | The status of the Focus View result or request. Possible values are:  
  - AVAILABLE: A Focus View has been generated and is available for viewing.  
  - DOES_NOT_EXIST: The Focus View requested on the device cannot be completed because the requested parameters do not exist on the device.  
  - PENDING: The Focus View has been requested.  
  - REQUEST: The Focus View has not been generated, but it can be requested.  
  - RETRY_REQUEST: The Focus View has not been generated. It was previously requested but no results were received. It can be requested again.  
  - UNAVAILABLE: The Focus View is not available, and the associated device is not online to fulfill the request. It can be requested at a later time.  
  - UNKNOWN_DEVICE: The Focus View is not available, and the associated device is no longer known. |
| relations  | A list of objects that are related to this Focus View. The following fields can be contained:  
  - Object: The URL of a Focus View, InstaQuery, or Detection Event that is linked to this Focus View.  
  - Relationship: How the relationship was established. |

**Search for Focus View Results**

Search for Focus Views by a list of Device ID and CylancePROTECT Event ID pairs, up to 200 at a time. The request requires both a CylancePROTECT Event ID and Device ID to determine whether or not a Focus View can be created.

**Service Endpoint:**

- `/foci/v2/search`

*For example:* [https://protectapi.cylance.com/foci/v2/search](https://protectapi.cylance.com/foci/v2/search)

**Method:**

- HTTP/1.1 POST

**Request Headers:**
- **Accept:** application/json
- **Authorization:** Bearer &lt;JWT Token returned by Auth API&gt; with the opticsfocus:list scope encoded

**Request**

```
[
  {
    "uid": "59F849F29BBE4F1F889AAF50F9153618",
    "device_id": "E378DACB9324453AB8C65A8406952195"
  }
]
```

**Response**

200 OK

```
[
  {
    "uid": "59F849F29BBE4F1F889AAF50F9153618",
    "device_id": "E378DACB9324453AB8C65A8406952195",
    "status": "PENDING",
    "focus_id": "A0AC3D2117C40D0576CED0D99069E96G"
  }
]
```

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such exception found.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

**Request JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uid</td>
<td>The unique ID of a CylancePROTECT event. Note: This is &quot;value&quot; from &quot;Get Focus View List&quot; on page 287.</td>
</tr>
<tr>
<td>device_id</td>
<td>The unique device ID that the lockdown command was issued to. Note: See &quot;About Device ID&quot; on page 25 for device ID formatting.</td>
</tr>
</tbody>
</table>

**Response JSON Schema Descriptions**
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uid</td>
<td>The unique ID of a CylancePROTECT event.</td>
</tr>
<tr>
<td>device_id</td>
<td>The unique device ID that the lockdown command was issued to.</td>
</tr>
<tr>
<td>status</td>
<td>The status of the Focus View result or request. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>- AVAILABLE: A Focus View has been generated and is available for viewing.</td>
</tr>
<tr>
<td></td>
<td>- PENDING: The Focus View has been requested.</td>
</tr>
<tr>
<td></td>
<td>- REQUEST: The Focus View has not been generated, but it can be requested.</td>
</tr>
<tr>
<td></td>
<td>- RETRY_REQUEST: The Focus View has not been generated. It was previously</td>
</tr>
<tr>
<td></td>
<td>requested but no results were received. It can be requested again.</td>
</tr>
<tr>
<td></td>
<td>- DOES_NOT_EXIST: The Focus View requested on the device cannot be</td>
</tr>
<tr>
<td></td>
<td>completed because the requested parameters do not exist on the device.</td>
</tr>
<tr>
<td></td>
<td>- UNAVAILABLE: The Focus View is not available, and the associated device is</td>
</tr>
<tr>
<td></td>
<td>not online to fulfill the request. It can be requested at a later time.</td>
</tr>
<tr>
<td></td>
<td>- UNKNOWN_DEVICE: The Focus View is not available, and the associated device</td>
</tr>
<tr>
<td></td>
<td>is no longer known.</td>
</tr>
<tr>
<td>focus_id</td>
<td>The unique ID of the Focus View.</td>
</tr>
</tbody>
</table>

### Request a Focus View

Request a Focus View from a specified device.

**Service Endpoint:**

```
/foci/v2
```

For example: https://protectapi.cylance.com/foci/v2

**Method:**

```
HTTP/1.1 POST
```

**Request Headers:**

```
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsfocus:create
  scope encoded
```

**Request**

```json
{
  "device_id": "E378DACB9324453AB8C65A8406952195",
```
"artifact_type": "Process",
"artifact_subtype": "Uid",
"value": "59F849F29BEB4F1F889AA50F9153618",
"threat_type": "THREAT",
"description": "Focus View Example"
}

Response

201 OK - Focus Created

{
  "device_id": "E378DACB9324453AB8C658406952195",
  "artifact_type": "Process",
  "artifact_subtype": "Uid",
  "value": "59F849F29BEB4F1F889AA50F9153618",
  "threat_type": "THREAT",
  "description": "Focus View Example",
  "id": "A0AC3D2117C405076C6D0990696E96C",
  "tenant_id": "4B1640D2D5631CF94A70DA1DCA6AA98",
  "created_at": "2018-07-26T01:20:07.596Z",
  "hostname": "User-Laptop-A123",
  "status": "AVAILABLE",
  "relations": [
    {
      "object": "/focus/focus_id",
      "relationship": "originated-from"
    }
  ]
}

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such exception found.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

Request JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>device_id</td>
<td>The unique device ID that the lockdown command was issued to.</td>
</tr>
<tr>
<td></td>
<td>Note: See &quot;About Device ID on page 25&quot; for device ID formatting.</td>
</tr>
<tr>
<td>artifact_</td>
<td>The type of Artifact for the Focus View.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| type       | - Protect: Request a Focus View for a CylancePROTECT-generated event.  
             - Process: Request a Focus View for a Process artifact to visualize how a process interacts with the device. This is the most common option.  
             - File: Request a Focus View for a File artifact to visualize how the file has been interacted with.  
             - NetworkConnection: Request a Focus View for a Network artifact to visualize communications associated with an IP address.  
             - RegistryKey: Request a Focus View for a Registry artifact to visualize how the registry key or path has been interacted with. |
| artifact_subtype | This field should always be "Uid" at this time. |
| value | The UID of the Artifact to gather a Focus View about. This can be obtained from InstaQuery results, another Focus View, the details/associated artifacts of a Detection Event, or anywhere else an Artifact is referenced. |
| threat_type | An option field to use with a "Protect" artifact_type to denote the type of threat that a Focus View is being generated for. |
| description | The human-readable description for the Focus View. |

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| device_id | The unique device ID that the lockdown command was issued to.  
**Note:** See "[About Device ID] on page 25" for device ID formatting. |
| artifact_type | The type of Artifact for the Focus View.  
- Protect: Request a Focus View for a CylancePROTECT-generated event.  
- Process: Request a Focus View for a Process artifact to visualize how a process interacts with the device. This is the most common option.  
- File: Request a Focus View for a File artifact to visualize how the file has been interacted with.  
- NetworkConnection: Request a Focus View for a Network artifact to visualize communications associated with an IP address.  
- RegistryKey: Request a Focus View for a Registry artifact to visualize how the registry key or path has been interacted with. |
<p>| artifact_subtype | This field should always be &quot;Uid&quot; at this time. |
| value | The UID of the Artifact to gather a Focus View about. This can be obtained from InstaQuery results, another Focus View, the details/associated artifacts of a Detection Event, or anywhere else an Artifact is referenced. |</p>
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>threat_type</td>
<td>An option field to use with a &quot;Protect&quot; artifact_type to denote the type of threat that a Focus View is being generated for.</td>
</tr>
<tr>
<td>description</td>
<td>The human-readable description for the Focus View.</td>
</tr>
<tr>
<td>id</td>
<td>The unique ID of the Focus View.</td>
</tr>
<tr>
<td>tenant_id</td>
<td>The unique ID of the tenant associated with the Focus View.</td>
</tr>
<tr>
<td>create_at</td>
<td>The timestamp (in UTC) of when the Focus View was created.</td>
</tr>
<tr>
<td>hostname</td>
<td>The hostname of the device that the Focus View was requested from.</td>
</tr>
<tr>
<td>status</td>
<td>The status of the Focus View result or request. Possible values are:</td>
</tr>
<tr>
<td></td>
<td>- AVAILABLE: A Focus View has been generated and is available for viewing.</td>
</tr>
<tr>
<td></td>
<td>- PENDING: The Focus View has been requested.</td>
</tr>
<tr>
<td></td>
<td>- REQUEST: The Focus View has not been generated, but it can be requested.</td>
</tr>
<tr>
<td></td>
<td>- RETRY_REQUEST: The Focus View has not been generated. It was previously requested but no results were received. It can be requested again.</td>
</tr>
<tr>
<td></td>
<td>- DOES_NOT_EXIST: The Focus View requested on the device cannot be completed because the requested parameters do not exist on the device.</td>
</tr>
<tr>
<td></td>
<td>- UNAVAILABLE: The Focus View is not available, and the associated device is not online to fulfill the request. It can be requested at a later time.</td>
</tr>
<tr>
<td></td>
<td>- UNKNOWN_DEVICE: The Focus View is not available, and the associated device is no longer known.</td>
</tr>
<tr>
<td>relations</td>
<td>A list of objects that are related to this Focus View. The following fields can be contained:</td>
</tr>
<tr>
<td></td>
<td>- Object: The URL of a Focus View, InstaQuery, or Detection Event that is linked to this Focus View.</td>
</tr>
<tr>
<td></td>
<td>- Relationship: How the relationship was established.</td>
</tr>
</tbody>
</table>

**Get a Focus View Summary**

Get the results of an existing Focus View.

**Service Endpoint:**

- `/foci/v2/{{focus_id}}`

For example:

https://protectapi.cylance.com/foci/v2/A0AC3D2117C40D0576CED0D99069E96G

**Method:**
HTTP/1.1 GET

Request Headers:
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsfocus:read scope encoded

Request
None

Response
200 OK

```
{
  "device_id": "E378DACB9324453AB8C65A8406952195",
  "artifact_type": "Process",
  "artifact_subtype": "Uid",
  "value": "59F849F29B8E4F1F889AAF50F9153618",
  "threat_type": "THREAT",
  "descriptions": "Focus View Example",
  "id": "A0AC3D2117C40D0576CED0D99069E96G",
  "tenant_id": "4B1640D2D56341CF94A70DA1DCA6AA98",
  "created_at": "2018-07-26T01:20:07.596Z",
  "hostname": "User-Laptop-A123",
  "status": "AVAILABLE",
  "relations": [
    {
      "object": "/survey/survey_id",
      "relationship": "origin-of"
    }
  ]
}
```

400 Bad Request - Malformed request.
401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.
403 Forbidden - The JWT Token did not contain the proper scope to perform this action.
404 Not Found - No such exception found.
500 Internal Server Error - An unforeseeable error has occurred.
503 Service Unavailable - Unable to respond at this time, please try again later.

Response JSON Schema Descriptions
<table>
<thead>
<tr>
<th><strong>Field Name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
</table>
| device_id     | The unique device ID that the lockdown command was issued to.  
**Note:** See "About Device ID" on page 25 for device ID formatting. |
| artifact_type | The type of Artifact for the Focus View.  
- Protect: Request a Focus View for a CylancePROTECT-generated event.  
- Process: Request a Focus View for a Process artifact to visualize how a process interacts with the device. This is the most common option.  
- File: Request a Focus View for a File artifact to visualize how the file has been interacted with.  
- NetworkConnection: Request a Focus View for a Network artifact to visualize communications associated with an IP address. |
| artifact_subtype | This field should always be "Uid" at this time. |
| value         | The UID of the Artifact used to gather the Focus View. |
| threat_type   | An option field to use with a "Protect" artifact_type to denote the type of threat that a Focus View is being generated for. |
| description   | The human-readable description for the Focus View. |
| id            | The unique ID of the Focus View. |
| tenant_id     | The unique tenant ID of the tenant that the device belongs to. |
| created_at    | The timestamp (in UTC) of when the file retrieval was requested. |
| hostname      | The hostname of the device that the file retrieval was requested on. |
| status        | The status of the Focus View result or request. Possible values are:  
- AVAILABLE: A Focus View has been generated and is available for viewing.  
- PENDING: The Focus View has been requested.  
- REQUEST: The Focus View has not been generated, but it can be requested.  
- RETRY_REQUEST: The Focus View has not been generated. It was previously requested but no results were received. It can be requested again.  
- DOES_NOT_EXIST: The Focus View requested on the device cannot be completed because the requested parameters do not exist on the device.  
- UNAVAILABLE: The Focus View is not available, and the associated device is not online to fulfill the request. It can be requested at a later time.  
- UNKNOWN_DEVICE: The Focus View is not available, and the associated device is no longer known. |
| relations     | A list of objects that are related to this Focus View. The following fields can be contained:  
- Object: The URL of a Focus View, InstaQuery, or Detection Event that is linked to this Focus View. |
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relationship</td>
<td>How the relationship was established.</td>
</tr>
</tbody>
</table>

## Get Focus View Results

Get the details of an existing Focus View that is used to generate the chart and table in the UI.

### Service Endpoint:

- `/foci/v2/{focus_id}/results`

For example:

https://protectapi.cylance.com/foci/v2/A0AC3D2117C40D0576CED0D99069E96G/results

### Method:

- HTTP/1.1 GET

### Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticsfocus:read scope encoded

### Request

None

### Response

200 OK

```json
{
  "id": "A0AC3D2117C40D0576CED0D99069E96G",
  "status": "AVAILABLE",
  "result": {}
}
```

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such exception found.

500 Internal Server Error - An unforeseeable error has occurred.
503 Service Unavailable - Unable to respond at this time, please try again later.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The unique ID of the Focus View.</td>
</tr>
</tbody>
</table>
| status     | The status of the Focus View result or request. Possible values are:  
  ■ AVAILABLE: A Focus View has been generated and is available for viewing.  
  ■ PENDING: The Focus View has been requested.  
  ■ REQUEST: The Focus View has not been generated, but it can be requested.  
  ■ RETRY_REQUEST: The Focus View has not been generated. It was previously requested but no results were received. It can be requested again.  
  ■ DOES_NOT_EXIST: The Focus View requested on the device cannot be completed because the requested parameters do not exist on the device.  
  ■ UNAVAILABLE: The Focus View is not available, and the associated device is not online to fulfill the request. It can be requested at a later time.  
  ■ UNKNOWN_DEVICE: The Focus View is not available, and the associated device is no longer known. |
| result     | The large structure of data that is used to generate the Focus View chart and table in the UI. This field will only be populated if the status field is AVAILABLE.  
  Parsing this data is beyond the scope of this guide. |
**InstaQuery**

The CylanceOPTICS InstaQuery API allows users to search for system artifacts stored locally by CylanceOPTICS - files, registry key persistence points, processes, etc. Users can investigate incidents, or hunt for potential threats, and then take appropriate remediation actions.

InstaQuery searches are zone based; unzoned endpoints cannot be searched via InstaQuery.

The CylanceOPTICS InstaQuery API includes:

- Creating an InstaQuery
- Getting a list of InstaQueries in a tenant
- Getting a specific InstaQuery
- Getting the results of an InstaQuery
- Archiving an InstaQuery

**Get InstaQueries**

Request a page with a list of CylanceOPTICS InstaQuery resources belonging to a tenant, sorted by occurrence time, in descending order (most recent occurred InstaQuery listed first).

The page number and page size parameters are optional, when the values are not specified, they default to 1 and 20 respectively.

**Service Endpoint:**

- `/instaqueries/v2?page=m&page_size=n`

For example: https://protectapi.cylance.com/instaqueries/v2?page=m&page_size=n

**Method:**

- HTTP/1.1 GET

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticssurvey:list scope encoded

**Request**

Append the following optional query string parameters:
- q: Case-insensitive search term (e.g. name, zones, artifact).
- archived: Include archived surveys.
- originated-from: Limit by the relationship.
- page: The page number to request. Defaults to 1.
- page_size: The number of detection records to retrieve per page. Defaults to 20.
- sort: Sort by field (adding '-' in front of the value denotes descending order).

Response

200 OK

```json
{
  "page_number": 1,
  "page_size": 20,
  "total_pages": 1,
  "total_number_of_items": 20,
  "page_items": [
    {
      "name": "InstaQuery Name",
      "description": "Test InstaQuery",
      "artifact": "File",
      "match_value_type": "Path",
      "match_values": [
        "exe"
      ],
      "case_sensitive": true,
      "match_type": "Fuzzy",
      "zones": [
        "D27FF5C95C0D4F56A00DA1FB297E440F"
      ],
      "filters": [
        {
          "aspect": "OS",
          "value": "Windows"
        }
      ],
      "relations": [
        {
          "object": "/focus/focus_id",
          "relationship": "originated-from"
        }
      ],
      "id": "AF593E38ED1B743BDC0A6FCC53A03CE",
      "archived": "2018-12-08T21:45:58-08:00",
      "results_available": true,
      "progress": {
        "queried": 500,
        "responded": 450
      }
    }
  ]
}
```
400 Bad Request - Returned for the following reasons:

- The tenant ID could not be retrieved from the JWT token specified in the Authorization header.
- The page number or page size specified are less than or equal to zero.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such exception found.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>page_number</td>
<td>The page number requested.</td>
</tr>
<tr>
<td>page_size</td>
<td>The page size requested.</td>
</tr>
<tr>
<td>total_number_of_items</td>
<td>The total number of resources.</td>
</tr>
<tr>
<td>total_pages</td>
<td>The total number of pages that can be retrieved based on the page size specified.</td>
</tr>
<tr>
<td>page_items</td>
<td>The list of detections belonging to the requested page.</td>
</tr>
<tr>
<td>name</td>
<td>The name of the InstaQuery.</td>
</tr>
<tr>
<td>description</td>
<td>The description of the InstaQuery.</td>
</tr>
<tr>
<td>artifact</td>
<td>The Artifact type that was queried.</td>
</tr>
<tr>
<td>match_value_type</td>
<td>The type (or Facet) of the Artifact that was queried.</td>
</tr>
<tr>
<td>match_values</td>
<td>The list of values that were queried for.</td>
</tr>
<tr>
<td>case_sensitive</td>
<td>Whether or not the InstaQuery should take case into account.</td>
</tr>
<tr>
<td>match_type</td>
<td>The match type configured for the query, either &quot;fuzzy&quot; or &quot;exact.&quot;</td>
</tr>
<tr>
<td>zones</td>
<td>The list of Zones queried.</td>
</tr>
<tr>
<td>filters</td>
<td>The list of filters applied to the InstaQuery.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>relations</td>
<td>The list of objects (e.g.: Focus Views) that the InstaQuery is related to.</td>
</tr>
<tr>
<td>id</td>
<td>The unique ID of the InstaQuery</td>
</tr>
<tr>
<td>archived</td>
<td>The timestamp of when the InstaQuery was archived.</td>
</tr>
<tr>
<td>results_available</td>
<td>Determines if the InstaQuery has returned any results.</td>
</tr>
<tr>
<td>progress</td>
<td>Provides the number of devices queried and the number of devices that have responded.</td>
</tr>
</tbody>
</table>

### Create InstaQuery

Update CylanceOPTICS InstaQuery resources for a specific tenant.

**Service Endpoint:**

- /instaqueries/v2

*For example:* https://protectapi.cylance.com/instaqueries/v2

**Method:**

- HTTP/1.1 POST

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticssurvey:create scope encoded

**Request**

```json
{
   "name": "InstaQuery Name",
   "description": "Test InstaQuery",
   "artifact": "File",
   "match_value_type": "Path",
   "match_values": [
      "exe"
   ],
   "case_sensitive": true,
   "match_type": "Fuzzy",
   "zones": [
      "D27FF5C45C0D4F56A00DA1FB297E440F"
   ],
   "filters": [
      {
         "aspect": "OS",
```
"value": "Windows"
],
"relations": [
{
"object": "/focus/focus_id",
"relationship": "originated-from"
}
]
}

Response

201 OK - Survey created

{
"name": "InstaQuery Name",
"description": "Test InstaQuery",
"artifact": "File",
"match_value_type": "Path",
"match_values": [
   "exe"
],
"case_sensitive": true,
"match_type": "Fuzzy",
"zones": [
   "D27FF5C45C0D4F56A00DA1FB297E440F"
],
"filters": [
{
"aspect": "OS",
"value": "Windows"
}
],
"relations": [
{
"object": "/focus/focus_id",
"relationship": "originated-from"
}
]
"id": "AF593F38EDC1B743BDC0A6FCC53A03CE",
"created_at": "2018-12-08T21:45:58-08:00",
"progress": {} 
}

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such exception found.
500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

**Request JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the InstaQuery.</td>
</tr>
<tr>
<td>description</td>
<td>The description of the InstaQuery.</td>
</tr>
<tr>
<td>artifact</td>
<td>The type of Artifact to search. Possible values are &quot;File&quot;, &quot;Process&quot;, &quot;NetworkConnection&quot;, and &quot;RegistryKey&quot;.</td>
</tr>
<tr>
<td>match_value_type</td>
<td>The type of value (also known as a Facet) to search. Possible values are dependent on the selected artifact type. Valid selections for each are as follows:</td>
</tr>
<tr>
<td></td>
<td>- File</td>
</tr>
<tr>
<td></td>
<td>- Path</td>
</tr>
<tr>
<td></td>
<td>- Md5</td>
</tr>
<tr>
<td></td>
<td>- Sha2</td>
</tr>
<tr>
<td></td>
<td>- Owner</td>
</tr>
<tr>
<td></td>
<td>- CreationDateTime</td>
</tr>
<tr>
<td></td>
<td>- Process</td>
</tr>
<tr>
<td></td>
<td>- Name</td>
</tr>
<tr>
<td></td>
<td>- Commandline</td>
</tr>
<tr>
<td></td>
<td>- PrimaryImagePath</td>
</tr>
<tr>
<td></td>
<td>- PrimaryImageMd5</td>
</tr>
<tr>
<td></td>
<td>- StartDateTime</td>
</tr>
<tr>
<td></td>
<td>- NetworkConnection</td>
</tr>
<tr>
<td></td>
<td>- DestAddr</td>
</tr>
<tr>
<td></td>
<td>- DestPort</td>
</tr>
<tr>
<td></td>
<td>- RegistryKey</td>
</tr>
<tr>
<td></td>
<td>- ProcessName</td>
</tr>
<tr>
<td></td>
<td>- ProcessPrimaryImagePath</td>
</tr>
<tr>
<td></td>
<td>- ValueName</td>
</tr>
<tr>
<td></td>
<td>- FilePath</td>
</tr>
<tr>
<td></td>
<td>- FileMd5</td>
</tr>
<tr>
<td></td>
<td>- IsPersistencePoint</td>
</tr>
<tr>
<td>match_values</td>
<td>A list of strings to be matched against for the InstaQuery.</td>
</tr>
<tr>
<td>case_sensitive</td>
<td>Determines whether to consider case sensitivity when matching values.</td>
</tr>
<tr>
<td>match_type</td>
<td>Determines whether or not to use an exact or &quot;fuzzy&quot; match. The default behavior of InstaQuery is to use a &quot;fuzzy&quot; match.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>name</td>
<td>The name of the InstaQuery.</td>
</tr>
<tr>
<td>description</td>
<td>The description of the InstaQuery.</td>
</tr>
<tr>
<td>artifact</td>
<td>The type of Artifact to search. Possible values are &quot;File&quot;, &quot;Process&quot;, &quot;NetworkConnection&quot;, and &quot;RegistryKey&quot;.</td>
</tr>
<tr>
<td>match_value_type</td>
<td>The type of value (also known as a Facet) to search. Possible values are dependent on the selected artifact type. Valid selections for each are as follows:</td>
</tr>
<tr>
<td></td>
<td>File</td>
</tr>
<tr>
<td></td>
<td>- Path</td>
</tr>
<tr>
<td></td>
<td>- Md5</td>
</tr>
<tr>
<td></td>
<td>- Sha2</td>
</tr>
<tr>
<td></td>
<td>- Owner</td>
</tr>
<tr>
<td></td>
<td>- CreationDateTime</td>
</tr>
<tr>
<td></td>
<td>Process</td>
</tr>
<tr>
<td></td>
<td>- Name</td>
</tr>
<tr>
<td></td>
<td>- CommandLine</td>
</tr>
<tr>
<td></td>
<td>- PrimaryImagePath</td>
</tr>
<tr>
<td></td>
<td>- PrimaryImageMd5</td>
</tr>
<tr>
<td></td>
<td>- StartDateTime</td>
</tr>
<tr>
<td></td>
<td>- NetworkConnection</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>match_values</td>
<td>A list of strings to be matched against for the InstaQuery.</td>
</tr>
<tr>
<td>case_sensitive</td>
<td>Determines whether to consider case sensitivity when matching values.</td>
</tr>
<tr>
<td>match_type</td>
<td>Determines whether or not to use an exact or &quot;fuzzy&quot; match. The default behavior of InstaQuery is to use a &quot;fuzzy&quot; match.</td>
</tr>
<tr>
<td>zones</td>
<td>A list of Zone IDs to perform the InstaQuery against.</td>
</tr>
<tr>
<td>filters</td>
<td>A list of filters when performing the InstaQuery.</td>
</tr>
<tr>
<td>aspect</td>
<td>The aspect (or type) of filters (e.g. &quot;OS&quot;).</td>
</tr>
<tr>
<td>value</td>
<td>The value to filter for (e.g. &quot;Windows&quot;).</td>
</tr>
<tr>
<td>relations</td>
<td>A list of objects (e.g. Focus View URLs) that are related to the InstaQuery. This is similar to the &quot;Pivot Query&quot; functionality in the Console.</td>
</tr>
<tr>
<td>object</td>
<td>The URL of the Focus View that the InstaQuery relates to.</td>
</tr>
<tr>
<td>relationship</td>
<td>How the InstaQuery relates to the URL. This should almost always be &quot;originated-from&quot;.</td>
</tr>
<tr>
<td>id</td>
<td>The unique identifier of the created InstaQuery.</td>
</tr>
<tr>
<td>created_at</td>
<td>The Date and Time that the InstaQuery was created.</td>
</tr>
<tr>
<td>progress</td>
<td>The progress of the InstaQuery.</td>
</tr>
</tbody>
</table>

**Get InstaQuery**

Request a specific InstaQuery resource belonging to a tenant.
Service Endpoint:

- /instaqueries/v2/{queryID}

For example:
https://protectapi.cylance.com/instaqueries/v2/AF593F38EDC1B743BDC0A6FCC53A03CE

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticssurvey:read scope encoded

Request

None

Response

200 OK

```
{
    "name": "InstaQuery Name",
    "description": "Test InstaQuery",
    "artifact": "File",
    "match_value_type": "Path",
    "match_values": [
        "exe"
    ],
    "case_sensitive": true,
    "match_type": "Fuzzy",
    "zones": [
        "D27FF5C45C0D4F56A00DA1FB297E440F"
    ],
    "filters": [
        {
            "aspect": "OS",
            "value": "Windows"
        }
    ],
    "relations": [
        {
            "object": "/focus/focus_id",
            "relationship": "originated-from"
        }
    ],
    "id": "AF593F38EDC1B743BDC0A6FCC53A03CE",
}
```
400 Bad Request - Returned for the following reasons:

- The Tenant ID could not be retrieved from the JWT token specified in the Authorization header.
- The unique identifier for the detection is not valid.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such exception found.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>The name of the InstaQuery.</td>
</tr>
<tr>
<td>description</td>
<td>The description of the InstaQuery.</td>
</tr>
<tr>
<td>artifact</td>
<td>The type of Artifact to search. Possible values are &quot;File&quot;, &quot;Process&quot;, &quot;NetworkConnection&quot;, and &quot;RegistryKey&quot;.</td>
</tr>
<tr>
<td>match_value_type</td>
<td>The type of value (also known as a Facet) to search. Possible values are dependent on the selected artifact type. Valid selections for each are as follows:</td>
</tr>
<tr>
<td></td>
<td>File</td>
</tr>
<tr>
<td></td>
<td>• Path</td>
</tr>
<tr>
<td></td>
<td>• MD5</td>
</tr>
<tr>
<td></td>
<td>• SHA256</td>
</tr>
<tr>
<td></td>
<td>• Owner</td>
</tr>
<tr>
<td></td>
<td>• CreationDateTime</td>
</tr>
<tr>
<td></td>
<td>Process</td>
</tr>
<tr>
<td></td>
<td>• Name</td>
</tr>
<tr>
<td></td>
<td>• CommandLine</td>
</tr>
<tr>
<td></td>
<td>• PrimaryImagePath</td>
</tr>
<tr>
<td></td>
<td>• PrimaryImageMd5</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>match_values</td>
<td>A list of strings to be matched against for the InstaQuery.</td>
</tr>
<tr>
<td>case_sensitive</td>
<td>Determines whether to consider case sensitivity when matching values.</td>
</tr>
<tr>
<td>match_type</td>
<td>Determines whether or not to use an exact or &quot;fuzzy&quot; match. The default behavior of InstaQuery is to use a &quot;fuzzy&quot; match. Possible values are: Fuzzy, Exact</td>
</tr>
<tr>
<td>zones</td>
<td>A list of Zone IDs to perform the InstaQuery against.</td>
</tr>
<tr>
<td>filters</td>
<td>A list of filters when performing the InstaQuery.</td>
</tr>
<tr>
<td>aspect</td>
<td>The aspect (or type) of filters (e.g. &quot;OS&quot;).</td>
</tr>
<tr>
<td>value</td>
<td>The value to filter for (e.g. &quot;Windows&quot;).</td>
</tr>
<tr>
<td>relations</td>
<td>A list of objects (e.g. Focus View URLs) that are related to the InstaQuery. This is similar to the &quot;Pivot Query&quot; functionality in the Console.</td>
</tr>
<tr>
<td>object</td>
<td>The URL of the Focus View that the InstaQuery relates to.</td>
</tr>
<tr>
<td>relationship</td>
<td>How the InstaQuery relates to the URL. This should almost always be &quot;originated-from&quot;.</td>
</tr>
<tr>
<td>id</td>
<td>The unique identifier of the created InstaQuery.</td>
</tr>
<tr>
<td>archived</td>
<td>The timestamp of when the InstaQuery was archived.</td>
</tr>
<tr>
<td>results_available</td>
<td>Determines if the InstaQuery has returned any results.</td>
</tr>
<tr>
<td>created_at</td>
<td>The Date and Time that the InstaQuery was created.</td>
</tr>
<tr>
<td>progress</td>
<td>The progress of the InstaQuery.</td>
</tr>
</tbody>
</table>
Get InstaQuery Results

Request a CylanceOPTICS InstaQuery resource results belonging to a tenant.

Service Endpoint:

- /instaqueries/v2(queryID)/results

For example:
https://protectapi.cylance.com/instaqueries/v2/AF593F38EDC1B743BDC0A6FCC53A03CE/results

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticssurvey:read scope encoded

Request

None

Response

200 OK

```json
{
    "id": "AF593F38EDC1B743BDC0A6FCC53A03CE",
    "status": "done",
    "result": [
        {
            "@timestamp": 1492623352.23335,
            "HostName": "User-Laptop-A123",
            "DeviceId": "E378DACB9324453AB8C65A8406952195",
            "@version": 1,
            "CorrelationId": "AF593F38EDC1B743BDC0A6FCC53A03CE",
            "Result": {
                "FirstObservedTime": "",
                "LastObservedTime": "",
                "Type": "File",
                "Uid": "",
                "Properties": {
                    "OwnerUid": "",
                    "SuspectedFileType": "",
                    "Size": 27136,
                    "Owner": "BUILTIN\Administrators",
                    "Path": "c:\windows\system32\svchost.exe"
                }
            }
        }
    ]
}
```
400 Bad Request - Returned for the following reasons:

- The Tenant ID could not be retrieved from the JWT token specified in the Authorization header.
- The unique identifier for the detection is not valid.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such exception found.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>The unique ID of the InstaQuery.</td>
</tr>
<tr>
<td>Status</td>
<td>The status of the InstaQuery.</td>
</tr>
<tr>
<td>Result</td>
<td>The list of responses to the InstaQuery.</td>
</tr>
<tr>
<td>@timestamp</td>
<td>The timestamp that the result was reported in Unix epoch time.</td>
</tr>
<tr>
<td>HostName</td>
<td>The hostname of the device that returned the result.</td>
</tr>
<tr>
<td>DeviceID</td>
<td>The unique ID of the device that returned the result.</td>
</tr>
<tr>
<td>@version</td>
<td>The version format of the result.</td>
</tr>
<tr>
<td>CorrelationID</td>
<td>The unique correlation ID of the result object.</td>
</tr>
<tr>
<td>Result</td>
<td>The object containing response data.</td>
</tr>
<tr>
<td>FirstObservedTime</td>
<td>The timestamp that the result was first observed on the system (e.g. when a File was first observed on the system as in a file being created).</td>
</tr>
<tr>
<td>LastObservedTime</td>
<td>The timestamp that the result was last observed on the system (e.g. when a File was last observed as in the last time a file was interacted with). <strong>Note:</strong> This value will be the same as the FirstObservedTimestamp for NetworkConnection and Process Artifacts.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Uid</td>
<td>The unique ID of the result.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of Artifact that the result's &quot;properties&quot; contain.</td>
</tr>
<tr>
<td>Properties</td>
<td>The object containing the individual elements of the result. This will vary depending on the Artifact and Type that was queried.</td>
</tr>
<tr>
<td></td>
<td><strong>File</strong></td>
</tr>
<tr>
<td></td>
<td>• Path: The full path to the file.</td>
</tr>
<tr>
<td></td>
<td>• CreationDateTime: The timestamp (in UTC) of when the file was created on the responding system.</td>
</tr>
<tr>
<td></td>
<td>• Md5: The MD5 hash of the file result (where applicable).</td>
</tr>
<tr>
<td></td>
<td>• Sha256: The SHA256 hash of the file result (where applicable).</td>
</tr>
<tr>
<td></td>
<td>• Owner: The owner of the file.</td>
</tr>
<tr>
<td></td>
<td>• SuspectedFileType: The suspected file type of the file object (where applicable).</td>
</tr>
<tr>
<td></td>
<td>• FileSignature: A set of information derived about the file's signature status.</td>
</tr>
<tr>
<td></td>
<td>• Size: The size of the file object (in bytes).</td>
</tr>
<tr>
<td></td>
<td>• OwnerUid: The unique ID of the owner of the file.</td>
</tr>
<tr>
<td></td>
<td><strong>Process</strong></td>
</tr>
<tr>
<td></td>
<td>• Name: The name of the process.</td>
</tr>
<tr>
<td></td>
<td>• CommandLine: The command line arguments that the process was executed with.</td>
</tr>
<tr>
<td></td>
<td>• StartDateTime: The timestamp (in UTC) of when the process was executed on the responding system.</td>
</tr>
<tr>
<td></td>
<td>• PrimaryImagePath: The image file path of the process.</td>
</tr>
<tr>
<td></td>
<td>• PrimaryImageMd5: The MD5 hash of the image file of the process.</td>
</tr>
<tr>
<td></td>
<td>• PrimaryImageSha256: The SHA256 hash of the image file of the process.</td>
</tr>
<tr>
<td></td>
<td>• PrimaryImageUid: The unique ID of the image file of the process.</td>
</tr>
<tr>
<td></td>
<td>• Owner: The user who owns the process.</td>
</tr>
<tr>
<td></td>
<td>• OwnerUid: The unique ID of the user who owns the process.</td>
</tr>
<tr>
<td></td>
<td>• SuspectedFileType: The suspected file type of the image file of the process.</td>
</tr>
<tr>
<td></td>
<td>• FileSignature: A set of information derived about the image file's signature status.</td>
</tr>
<tr>
<td></td>
<td>• IsBeingDebugged: A Boolean value to determine if the process has a debugger attached to it.</td>
</tr>
<tr>
<td></td>
<td><strong>Network</strong></td>
</tr>
<tr>
<td></td>
<td>• DestinationAddress: The IP address that the connection was destined to.</td>
</tr>
<tr>
<td></td>
<td>• DestinationPort: The port associated with the remote IP address.</td>
</tr>
</tbody>
</table>
### Field Name | Description
--- | ---
- ProcessName: The process name that was associated with the connection.  
- ProcessPrimaryImageUid: The unique ID of the process associated with the connection.  
- ProcessPrimaryImagePath: The image file path of the process associated with the connection.  
- ProcessImageMd5: The MD5 hash of the image file of the process associated with the connection.  
- ProcessImageSha256: The SHA256 hash of the image file of the process associated with the connection.  
- SuspectedFileType: The suspected file type of the image file of the process associated with the connection.  
- Registry:  
  - IsPersistencePoint: A binary value (1 or 0) to determine if the resulting Registry item is a common persistence location.  
  - ValueName: The name of the Registry Value that was interacted with.  
  - Path: The full path of the Registry Key.  
  - FilePath: The full path of the file referenced in the Registry Value (where applicable).  
  - FileMd5: The MD5 hash of the file referenced in the Registry Value (where applicable).  
  - FileSha256: The SHA256 hash of the file referenced in the Registry Value (where applicable).  
  - FileUid: The unique ID of the file referenced in the Registry Value (where applicable).  
  - SuspectedFileType: The suspected file type of the file referenced in the Registry Value (where applicable).  
  - FileSignature: A set of information derived about a file's signature status that is referenced in the Registry Value (where applicable).  

### Archive InstaQuery

Archive a CylanceOPTICS InstaQuery resource belonging to a tenant. Surveys are archived instead of deleted so that user activity history can be maintained.

**Service Endpoint:**

`/instaqueries/v2/{queryID}/archive`

For example:

```
https://protectapi.cylance.com/instaqueries/v2/AF593F38EDC1B743BDC0A6FCC53A03CE/archive
```
Method:
- HTTP/1.1 POST

Request Headers:
- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticssurvey:update scope encoded

Request
None

Response
200 OK - Survey archived.

400 Bad Request - Returned for the following reasons:
- The Tenant ID could not be retrieved from the JWT token specified in the Authorization header.
- The unique identifier for the detection is not valid.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - No such exception found.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.
CylanceOPTICS Policy

The CylanceOPTICS Policy API allows administrators to get a list of all policies and if a Detection Rule Set is assigned to each policy. Administrators can also assign or unassign a Detection Rule Set to a policy.

Get Detection Rule Sets to Policy Mapping

Get a list of Cylance policies, the unique ID of the Detection Rule Set currently assigned to the policy, and a list of all Detection Rule Sets available to the policy.

Service Endpoint:

- /opticsPolicies/v2/configurations

For example:

https://protectapi.cylance.com/opticsPolicies/v2/configurations

Method:

- HTTP/1.1 GET

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticspolicy:list scope encoded

Request:

None

Response

200 OK - A list of policies and Detection Rule Sets display.

```json
{
    "page_size": 10,
    "total_pages": 1,
    "page_items": [
        {
            "policy_id": "d5c6d6a3-0599-4fb5-96bc-0f0dc7eac6ea",
            "policy_configurations": {
                "DETECTION": "00000000-0000-0000-0000-000000000000"
            },
            "device_count": 2,
            "configurations": [
                {
                    "type": "DETECTION",
```
400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resources page requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>configuration_id</td>
<td>The unique ID for the Detection Rule Set. Matching this number to the DETECTION number gives you the name of the Detection Rule Set assigned to the policy.</td>
</tr>
</tbody>
</table>
## Get Detection Rule Set for a Policy

Get the Detection Rule Set assigned to a policy.

**Service Endpoint:**

- `/opticsPolicies/v2/configurations/{policy_id}`

For example:

https://protectapi.cylance.com/opticsPolicies/v2/configurations/d5c6d6a3-0599-4fb5-96bc-0fdc7eacb6ea

**Method:**

- HTTP/1.1 GET

**Request Headers:**

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticspolicy:read scope encoded

**Request:**

None

**Response**

200 OK - A list of policies and Detection Rule Sets display.


```
{
    "policy_id": "d5c6d6a3-0599-4fb5-96bc-0fcd7eacb6ea",
    "policy_configurations": {
        "DETECTION": "00000000-0000-0000-0000-000000000000"
    },
    "configurations": [
        {
            "type": "DETECTION",
            "display_name": "(None)",
            "configuration_id": "00000000-0000-0000-0000-000000000000"
        },
        {
            "type": "DETECTION",
            "display_name": "(None)",
            "configuration_id": "d23198bd-2725-4660-969c-971f1548ffc3"
        }
    ],
    "device_count": 100
}
```

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resources page requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

**Response JSON Schema Descriptions**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>configuration_id</td>
<td>The unique ID for the Detection Rule Set. Matching this number to the DETECTION number gives you the name of the Detection Rule Set assigned to the policy.</td>
</tr>
<tr>
<td>DETECTION</td>
<td>The unique ID for the Detection Rule Set assigned to the policy.</td>
</tr>
<tr>
<td>device_count</td>
<td>The number of devices assigned to the policy.</td>
</tr>
<tr>
<td>display_name</td>
<td>The Detection Rule Set name.</td>
</tr>
<tr>
<td>policy_id</td>
<td>The unique ID for the policy.</td>
</tr>
<tr>
<td>type</td>
<td>The configuration type. For Detection Rule Sets, this is DETECTION.</td>
</tr>
</tbody>
</table>

**Update a Detection Rule Set in a Policy**

Update the Detection Rule Set assigned to a policy.
Service Endpoint:

- /opticsPolicies/v2/configurations

For example:
https://protectapi.cylance.com/opticsPolicies/v2/configurations

Method:

- HTTP/1.1 POST

Request Headers:

- Accept: application/json
- Authorization: Bearer <JWT Token returned by Auth API> with the opticspolicy:create scope encoded

Request

```json
{
    "configuration_type": "DETECTION",
    "configuration_id": "d23198bd-2725-4660-969c-971f1548ffc3",
    "link": ["d5c6d6a3-0599-4fb5-96bc-0fdec7eacb6ea"],
    "unlink": []
}
```

Response

200 OK - The Detection Rule Set was updated in the policy.

```
[
    {
        "id": "d5c6d6a3-0599-4fb5-96bc-0fdec7eacb6ea",
        "success": "true",
        "message": "link",
    }
]
```

400 Bad Request - Malformed request.

401 Unauthorized - The JWT Token is not specified, has expired, or is otherwise invalid.

403 Forbidden - The JWT Token did not contain the proper scope to perform this action.

404 Not Found - The detection resources page requested doesn't exist.

500 Internal Server Error - An unforeseeable error has occurred.

503 Service Unavailable - Unable to respond at this time, please try again later.

Request JSON Schema Descriptions
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>configuration_id</td>
<td>The Detection Rule Set unique identifier (ID). Use Get All Detection Rule Sets or Get Single Detection Rule Set requests to get this ID.</td>
</tr>
<tr>
<td>configuration_type</td>
<td>The configuration type. For Detection Rule Sets, this is DETECTION.</td>
</tr>
<tr>
<td>link</td>
<td>Add the Policy ID to assign the Detection Rule Set to the policy.</td>
</tr>
<tr>
<td>unlink</td>
<td>Add the Policy ID to remove the Detection Rule Set from the policy.</td>
</tr>
</tbody>
</table>

### Response JSON Schema Descriptions

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>The Policy ID.</td>
</tr>
<tr>
<td>message</td>
<td>Displays the assignment of the Detection Rule Set to the policy.</td>
</tr>
<tr>
<td></td>
<td>- link: The Detection Rule Set is assigned to the policy.</td>
</tr>
<tr>
<td></td>
<td>- unlink: The Detection Rule Set was removed from the policy.</td>
</tr>
<tr>
<td>success</td>
<td>Displays if the update was successful or not.</td>
</tr>
<tr>
<td></td>
<td>- false: Updating the Detection Rule Set in the policy failed.</td>
</tr>
<tr>
<td></td>
<td>- true: Updating the Detection Rule Set in the policy succeeded.</td>
</tr>
</tbody>
</table>
The following is information about some REST and JSON tools that might help you when using the Cylance APIs.

**Note:** Cylance supports Cylance API resources, including helping users troubleshoot Cylance API requests. Cylance does not write or train users on how to create scripts or code (like using Python).

## REST Clients

Although the intent of the Cylance API is to facilitate easy integration of Cylance and other systems through the organization's developed code, using or testing the Cylance API doesn't require any specific programming knowledge. Free tools are available for download that allow you to make ad hoc REST requests to the Cylance API. Some examples are:

- **Fiddler** ([https://www.telerik.com/fiddler](https://www.telerik.com/fiddler)) - Free web debugging proxy. Also has an easy-to-use composer and replay features for HTTP requests.

- **Postman** ([https://www.getpostman.com](https://www.getpostman.com)) - Google Chrome browser extension designed for testing REST APIs. There are also native Windows, macOS, and Linux clients available.

## JSON Validators

Cylance API requests and responses use JSON for the body payload. If the body used in the request doesn't conform to proper JSON formatting, it will result in an HTTP response of 400 - Bad Request. To ensure that your JSON is properly formatted, use one of these free, popular tools:

- **JSON Formatter and Validator** ([http://jsonformatter.curiousconcept.com](http://jsonformatter.curiousconcept.com)) - Online, simple interface with options to define and transform the output according to the desired level of white space. Provides highlights and informative descriptions of errors.

- **Notepad++** ([https://notepad-plus-plus.org](https://notepad-plus-plus.org)) - Freeware text editor. Supports a wide variety of plug-in extensions, including various JSON formatting and validation tools (like JSTool and JSON Viewer).
Appendix
Find File Checksum

When uploading a Package, the SHA256 hash is required for the checksum.

Windows

CertUtil is a pre-installed Windows utility. This utility can provide the hash checksum for a file.

1. Open the **Command Prompt**. For Windows 10, click the Start menu, type `cmd`, then click Command Prompt.

2. Type `CertUtil -hashfile pathoffile hashtype`, then press **Enter**.
   a. Replace `<pathoffile>` with the path to the file to check. Example: `c:\test\hello_world.py`.
   b. Replace `<hashtype>` with the hash format. Example: SHA256. Valid hash formats include: MD2, MD4, MD5, SHA1, SHA256, SHA384, and SHA512.
   c. Example command: `CertUtil -hashfile c:\test\hello_world.py SHA256`

3. Copy the SHA256 hash.

Figure - Use CertUtil to Find Checksum

macOS

1. Open **Terminal**. Search for Terminal in Spotlight or Launchpad, or open using **Applications > Utilities**.

2. Type `shasum -a 256 <pathoffile>`, then press **Return**.

3. Copy the SHA256 hash.

Linux

1. Open **Terminal**.

2. Type `sha256sum <pathoffile>`, then press **Enter**.

3. Copy the SHA256 hash.

Threat Classifications

In the Cylance Console, there exists classification information for threats reported in your organization.
The following is a list of possible file status entries that may appear under classification for each threat, along with a brief description of each entry.

**File Unavailable**

Due to an upload constraint (example: file is too large to upload), the file is unavailable for analysis. If classification is necessary, contact Cylance Support for an alternate method to transfer the file for analysis.

Unknown (blank entry)

The file has not been analyzed by the Cylance Research team. Once the file is analyzed, the classification will be updated with a new status.

**Trusted - Local**

The file has been analyzed by the Cylance Research team and has been deemed safe (not malicious, not a PUP). A file identified as Trusted - Local can be globally safe listed so that the file will be allowed to execute and not generate any additional alerts if found on other devices within your organization. The reason for the "Local" designation is due to the fact that the file did not come from a trusted source (such as Microsoft or other trusted installers) and therefore cannot be added to our trusted cloud repository.

**PUP**

The file has been identified as a Potentially Unwanted Program (PUP). This indicates that the program may be unwanted, despite the possibility that users consented to download it. Some PUPs may be permitted to run on a limited set of systems in your organization (example: a VNC application allowed to run on Domain Admin devices). A Cylance Console administrator can choose to waive or block PUPs on a per device basis or globally quarantine or safe list the file based on company policies. Depending on how much analysis can be performed against a PUP, further subclassification may be possible. Those subclasses are shown below and will aid an administrator in determining whether a particular PUP should be blocked or allowed to run.

<table>
<thead>
<tr>
<th>Subclass</th>
<th>Definition</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adware</td>
<td>Technologies that provide annoying advertisements (example: pop-ups) or provide bundled third-party add-ons when installing an application. This usually occurs without adequate notification to the user about the nature or presence of the add-on, control over installation, control over use, or the ability to fully uninstall the add-on.</td>
<td>Gator, Adware Info</td>
</tr>
<tr>
<td>Corrupt</td>
<td>Any executable that is malformed and unable to run.</td>
<td></td>
</tr>
<tr>
<td>Subclass</td>
<td>Definition</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Game</td>
<td>Technologies that create an interactive environment with which a player can play.</td>
<td>Steam Games, League of Legends</td>
</tr>
<tr>
<td>Generic</td>
<td>Any PUP that does not fit into an existing category.</td>
<td></td>
</tr>
<tr>
<td>HackingTool</td>
<td>Technologies that are designed to assist hacking attempts.</td>
<td>Cobalt Strike, MetaSpl0it</td>
</tr>
<tr>
<td>Portable Application</td>
<td>Program designed to run on a computer independently, without needing installation.</td>
<td>Turbo</td>
</tr>
<tr>
<td>Scripting Tool</td>
<td>Any script that is able to run as if it were an executable.</td>
<td>AutoIT, py2exe</td>
</tr>
<tr>
<td>Toolbar</td>
<td>Technologies that place additional buttons or input boxes on-screen within a UI.</td>
<td>Nasdaq Toolbar, Bring Me Sports</td>
</tr>
<tr>
<td>Other</td>
<td>Is a category for things that don't fit anything else, but are still PUP's. There are a lot of different PUP's, most of which are not malicious but serveral that should still be brought to the attention of the System Administrators through our product. Usually because they have potentially negative uses or negatively impact a system or network.</td>
<td></td>
</tr>
</tbody>
</table>

### Dual Use

Dual Use indicates the file can be used for malicious and non-malicious purposes. Caution should be used when allowing the use of these files in your organization.

<table>
<thead>
<tr>
<th>Subclass</th>
<th>Definition</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crack</td>
<td>Technologies that can alter (or crack) another application in order to bypass licensing limitations or Digital Rights Management (DRM) protection.</td>
<td></td>
</tr>
<tr>
<td>Generic</td>
<td>Any Dual Use tool that does not fit into an existing subclass.</td>
<td></td>
</tr>
<tr>
<td>KeyGen</td>
<td>Technologies which can generate or recover/reveal product keys that can be used to bypass Digital Rights Management (DRM) or licensing protection of software and other digital media.</td>
<td></td>
</tr>
<tr>
<td>MonitoringTool</td>
<td>Technologies that track a user's online activities without awareness of the user by logging and possibly transmitting logs of one or more of the following:</td>
<td>Veriato 360, Refog Keylogger</td>
</tr>
</tbody>
</table>
### Malware

The Cylance Research team has definitively identified the file as a piece of malware; the file should be removed or quarantined as soon as possible. Verified malware can be further subclassified.

<table>
<thead>
<tr>
<th>Subclass</th>
<th>Definition</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backdoor</td>
<td>Malware that provides unauthorized access to a system, bypassing security measures.</td>
<td>Back Orifice, Eleanor</td>
</tr>
<tr>
<td>Bot</td>
<td>Malware that connects to a central Command and Control (C&amp;C) botnet server.</td>
<td>QBot, Koobface</td>
</tr>
<tr>
<td>Downloader</td>
<td>Malware that downloads data to the host system.</td>
<td>Staged-Downloader</td>
</tr>
<tr>
<td>Dropper</td>
<td>Malware that installs other malware on a system.</td>
<td></td>
</tr>
<tr>
<td>Exploit</td>
<td>Malware that attacks a specific vulnerability on the system.</td>
<td></td>
</tr>
<tr>
<td>FakeAlert</td>
<td>Malware that masquerades as legitimate security software to trick the user into fixing fake security problems at a price.</td>
<td>Fake AV White Paper</td>
</tr>
<tr>
<td>Generic</td>
<td>Any malware that does not fit into an existing category.</td>
<td></td>
</tr>
<tr>
<td>InfoStealer</td>
<td>Malware that records login credentials and/or other sensitive information.</td>
<td>Snifula</td>
</tr>
<tr>
<td>Parasitic</td>
<td>Parasitic viruses, also known as file viruses, spread by attaching</td>
<td></td>
</tr>
</tbody>
</table>
Subclass | Definition | Examples
--- | --- | ---
Ransom | Malware that restricts access to system or files and demands payment for removal of restriction, thereby holding the system for ransom. | CryptoLocker, CryptoWall
Remnant | Any file that has malware remnants post removal attempts. |
Rootkit | Malware that enables access to a computer while shielding itself or other files to avoid detection and/or removal by administrators or security technologies. | TDL, Zero Access Rootkit
Trojan | Malware that disguises itself as a legitimate program or file. | Zeus
Virus | Malware that propagates by inserting or appending itself to other files. | Sality, Virut
Worm | Malware that propagates by copying itself to another device. | Code Red, Stuxnet

Scope Values for Authentication Token

The following are scope values and descriptions related to the Authentication Token. See "Authentication and Authorization" on page 19.

<table>
<thead>
<tr>
<th>Scope</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>application:list</td>
<td></td>
</tr>
<tr>
<td>device:delete</td>
<td></td>
</tr>
<tr>
<td>device:list</td>
<td></td>
</tr>
</tbody>
</table>

Custom Rule Guide

The CylanceOPTICS Context Analysis Engine Custom Rule Builder allows users to extend the logic of behavioral rules provided by Cylance as well as the ability to create their own logic to detect malicious or suspicious behaviors in their own environments.

**Note:** User imported Custom Rules are not supported by Cylance Support.

The Context Analysis Engine (CAE) rules consist of five primary pieces of data:

- **States:** States define the flow of a CAE Rule. These allow CylanceOPTICS to statefully observe a series of Events that occur on a device. These represent a "1, then 2, then"
3rd scenario that might occur.

- **Functions**: Functions define the logic required to successfully fulfill a State. This logic applies directly to the defined Field Operators and is used to represent a "A, and B, and C" or "A, and B, but not C" attributes of an Event that occurs on a device.

- **Field Operators**: Field Operators define how Operands (Facet Value Extractors) are evaluated. Field Operators include actions like Equals, Contains, and Is True.

- **Operands (Facet Value Extractors)**: Operands act as the values being compared by CylanceOPTICS. Operands allow extracting specific pieces of data about an Event on a device (like File Paths, File Hashes, and Process Names) and compare those with literal values (like String, Decimal, Boolean, and Integer).

- **Artifacts of Interest (AOI)**: AOI define the points where CylanceOPTICS can interact with a Rule to take automated response actions. These artifacts are targeted by CylanceOPTICS when conducting actions such as Terminating Processes, Logging Off Users, or Deleting Files.

**Sample Rule**

```json
{
   "States": [
      {
         "Name": "TestFile",
         "Scope": "Global",
         "Function": "(a)",
         "FieldOperators": {
            "a": {
               "Type": "Contains",
               "Operands": [
                  {
                     "Source": "TargetFile",
                     "Data": "Path"
                  },
                  {
                     "Source": "Literal",
                     "Data": "my_test_file"
                  }
               
            }
         
            "OperandType": "String"
      }
      
      "ActivationTimeLimit": "-0:00:00.001",
      "Actions": [
         
         "Type": "AOI",
         "ItemName": "InstigatingProcess",
         "Position": "PostActivation"
      
      }
   }]
}```
States

States are the highest logic level of a rule and have a larger number of required fields.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actions</td>
<td>Contains a list of objects used to define Artifacts of Interest within a state. The details about AOI are provided in a later section.</td>
</tr>
<tr>
<td>ActivationTimeLimit</td>
<td>Defines how long CylanceOPTICS will wait for events to trigger the event. This should be the default value of -0:00:00:001.</td>
</tr>
<tr>
<td>FieldOperators</td>
<td>An object that contains the field operators and operands that should be inspected to fulfill the Function defined in the State. The details of this are provided in a later section.</td>
</tr>
<tr>
<td>Filters</td>
<td>Defines which Event Categories, Subcategories, and Types that CylanceOPTICS should inspect when attempting to fulfill a State. The details of this are provided in a later section.</td>
</tr>
<tr>
<td>Function</td>
<td>Contains the logic function that CylanceOPTICS must observe to consider a State as satisfied. The details of this are provided in a later section.</td>
</tr>
</tbody>
</table>
### Field

Functions define the logic required to successfully fulfill a State. This logic applies directly to the defined Field Operators and is used to represent an A, and B, and C or A, and B, but not C attributes of an event that occurs on a device. This logic applies directly to the defined Field Operators within a State.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND - &amp;</td>
<td>Requires that two or more Field Operators be matched to consider the State satisfied.</td>
<td>a &amp; b &amp; c</td>
</tr>
<tr>
<td>OR -</td>
<td>Requires that one of two or more Field Operators be matched to consider the State satisfied.</td>
<td>a</td>
</tr>
<tr>
<td>NOT - !</td>
<td>Requires that a defined Field Operator be False or Not Matched to consider the State satisfied.</td>
<td>a &amp; b &amp; !c</td>
</tr>
<tr>
<td>GROUP - ()</td>
<td>Groups a set of Field Operators together to fulfill more complex logic requirements.</td>
<td>(a &amp; b)</td>
</tr>
</tbody>
</table>

### Field Operators

Field Operators are the logical pieces of a rule that allow CylanceOPTICS to compare two values. If there are two or more Operands, and they match the comparison criteria, CylanceOPTICS will consider that portion of the defined Function as complete. When all pieces of the Function are complete, the State will be satisfied.

The Field Operators field is an object that will consist of one or more conditional objects. These conditional objects can be set to any value, however, they must match the same conditional values that are referenced in the Function field. As such, Cylance recommends that these names are kept to simple and logical values, such as numbers or letters.
<table>
<thead>
<tr>
<th>Field Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContainsAll</td>
<td>Determines if the specified operand contains all of the operands from a set. Positive: &quot;hello, I am a string&quot; contains all from (&quot;ello&quot;, &quot;ng&quot;). Negative: &quot;hello, I am a string&quot; does not contain all from (&quot;hi&quot;, &quot;ng&quot;).</td>
</tr>
<tr>
<td>ContainsAllWords</td>
<td>Determines if the specified operand contains all of the operands from a set, where each set operand must appear as a whole word surrounded by white space, punctuation, or end/beginning string markers. Positive: &quot;hello, I am a string&quot; contains all words from (&quot;hello&quot;, &quot;a&quot;, &quot;string&quot;). Negative: &quot;hello, I am a string&quot; does not contain all words from (&quot;ello&quot;, &quot;ng&quot;).</td>
</tr>
<tr>
<td>Contains</td>
<td>Determines if the specified operand contains any of the operands from a set. Positive: &quot;hello, I am a string&quot; contains any from (&quot;ello&quot;, &quot;banana&quot;). Negative: &quot;hello, I am a string&quot; does not contain any from (&quot;hi&quot;, &quot;banana&quot;).</td>
</tr>
<tr>
<td>ContainsWord</td>
<td>Determines if the specified operand contains any of the operands from a set, where each set operand would have to appear as a whole word surrounded by white space, punctuation, or end/beginning string markers. Positive: &quot;hello, I am a string&quot; contains any words from (&quot;hello&quot;, &quot;banana&quot;). Negative: &quot;hello, I am a string&quot; does not contain any words from (&quot;ello&quot;, &quot;ng&quot;).</td>
</tr>
<tr>
<td>EndsWith</td>
<td>Determines if the specified left operand ends with the specified right operand. Positive: &quot;hello, I am a string&quot; ends with &quot;ring&quot;. Negative: &quot;hello, I am a string&quot; does not end with &quot;bring&quot;.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines if the specified operand equals exactly any of the operands from a set, where each set operand would have to appear as a number or a whole word surrounded by white space, punctuation, or end/beginning string markers. Positive: 10 equals any from (10, 20, 30). Positive: &quot;hello&quot; equals any from (&quot;hello&quot;, &quot;banana&quot;). Negative: 100 does not equal any from (10, 20, 30). Negative: &quot;hello&quot; does not equal any from (&quot;ello&quot;, &quot;ng&quot;).</td>
</tr>
<tr>
<td>GreaterThan</td>
<td>Determines if the specified left operand is greater than the specified right operand. Positive: 14.4 is greater than 10.1. Negative: 1 is not greater than 1000.</td>
</tr>
<tr>
<td>GreaterThanOrEquals</td>
<td>Determines if the specified left operand is greater than or equal to the</td>
</tr>
<tr>
<td>Field Operator</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>specified right operand.</td>
<td></td>
</tr>
<tr>
<td>Positive: 14.4 is greater than or equal to 10.1.</td>
<td></td>
</tr>
<tr>
<td>Negative: 1 is not greater than or equal to 1000.</td>
<td></td>
</tr>
<tr>
<td>InRange</td>
<td>Determines if the specified middle operand is between the left and right operands.</td>
</tr>
<tr>
<td>Positive: 10 is between 1 and 20.</td>
<td></td>
</tr>
<tr>
<td>Positive: 5.3 is between 5.3 and 20.1 (inclusive).</td>
<td></td>
</tr>
<tr>
<td>Negative: 4 is not between 5 and 10.</td>
<td></td>
</tr>
<tr>
<td>Negative: 20 is not between 20 and 40 (exclusive).</td>
<td></td>
</tr>
<tr>
<td>InPsInRange</td>
<td>Determines if the TargetNetworkConnection address (SourceAddress, DestinationAddress) is within the specified &quot;min&quot; and &quot;max&quot; options.</td>
</tr>
<tr>
<td>Allowed Operands are:</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>&quot;Source&quot;: &quot;TargetNetworkConnection&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;Data&quot;: &quot;SourceAddress&quot;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>And:</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>&quot;Source&quot;: &quot;TargetNetworkConnection&quot;,</td>
<td></td>
</tr>
<tr>
<td>&quot;Data&quot;: &quot;DestinationAddress&quot;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>IsHomoglyph</td>
<td>Determines if the left operand is a homoglyph of the right operand. Homoglyphs are things that appear to have the same meaning visually, but are actually different.</td>
</tr>
<tr>
<td>For example, a US Latin 1 &quot;e&quot; and a French &quot;e&quot; appear to be the same character and have the same meaning, but the computer sees them as different values.</td>
<td></td>
</tr>
<tr>
<td>Positive: &quot;3xplor3&quot; is a homoglyph of &quot;explore&quot; with 100% certainty.</td>
<td></td>
</tr>
<tr>
<td>Positive: &quot;3xplord&quot; is a homoglyph of &quot;explore&quot; with 90% certainty.</td>
<td></td>
</tr>
<tr>
<td>Negative: &quot;temp&quot; is not a homoglyph of &quot;temp&quot; because these are the same string.</td>
<td></td>
</tr>
<tr>
<td>Negative: &quot;431&quot; is not a homoglyph of &quot;big&quot; because these share no transitive characteristics.</td>
<td></td>
</tr>
<tr>
<td>IsNullOrEmpty</td>
<td>Determines if the specified operand is null or empty.</td>
</tr>
<tr>
<td>Positive: &lt;null&gt; is null or empty.</td>
<td></td>
</tr>
<tr>
<td>Positive: &quot;&quot; is null or empty.</td>
<td></td>
</tr>
<tr>
<td>Positive: &quot; &quot; is null or empty.</td>
<td></td>
</tr>
<tr>
<td>Negative: &quot;Hello&quot; is not null or empty.</td>
<td></td>
</tr>
<tr>
<td>Field Operator</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IsPopulated</td>
<td>Determines if the specified operand is not null or empty.</td>
</tr>
<tr>
<td></td>
<td>Positive: &quot;Hello&quot; is not null or empty.</td>
</tr>
<tr>
<td></td>
<td>Negative: &lt;null&gt; is null or empty.</td>
</tr>
<tr>
<td></td>
<td>Negative: &quot;&quot; is null or empty.</td>
</tr>
<tr>
<td></td>
<td>Negative: &quot; &quot; is null or empty.</td>
</tr>
<tr>
<td>IsTrue</td>
<td>Determines if the specified value is True.</td>
</tr>
<tr>
<td></td>
<td>Positive: TriState.True</td>
</tr>
<tr>
<td></td>
<td>Negative: TriState.False</td>
</tr>
<tr>
<td></td>
<td>Negative: TriState.Unknown</td>
</tr>
<tr>
<td>LessThan</td>
<td>Determines if the specified left operand is less than the specified right operand.</td>
</tr>
<tr>
<td></td>
<td>Positive: 4.4 is less than 10.1.</td>
</tr>
<tr>
<td></td>
<td>Negative: 1000 is not less than 1.</td>
</tr>
<tr>
<td>LessThanOrEquals</td>
<td>Determines if the specified left operand is less than or equal to the specified right operand.</td>
</tr>
<tr>
<td></td>
<td>Positive: 4.4 is less than or equal to 10.1.</td>
</tr>
<tr>
<td></td>
<td>Positive: 14 is less than or equal to 14.</td>
</tr>
<tr>
<td></td>
<td>Negative: 1000 is not less than or equal to 1.</td>
</tr>
<tr>
<td>LevenshteinDistance</td>
<td>Determines if the distance, the number of changes needed to turn one operand into another operand, is within an acceptable range.</td>
</tr>
<tr>
<td></td>
<td>Positive: &quot;cat&quot; is within a Levenshtein Distance of 1 from &quot;bat&quot;.</td>
</tr>
<tr>
<td></td>
<td>Positive: &quot;hello&quot; is within a Levenshtein Distance of 3 from &quot;bell&quot;.</td>
</tr>
<tr>
<td></td>
<td>Negative: &quot;cart&quot; is not within a Levenshtein Distance of 1 from &quot;act&quot;.</td>
</tr>
<tr>
<td>RegexMatches</td>
<td>Determines if the specified operand conforms to a regular expression.</td>
</tr>
<tr>
<td></td>
<td>Positive: &quot;hello, I am a string&quot; conforms to &quot;&quot;hello, [li] am [aA] string$&quot;.</td>
</tr>
<tr>
<td></td>
<td>Negative: &quot;hello, I am a string&quot; does not conform to &quot;[hi]hey, I am a string$&quot;.</td>
</tr>
<tr>
<td>StartsWith</td>
<td>Determines if the specified left operand starts with the specified right operand.</td>
</tr>
<tr>
<td></td>
<td>POSITIVE: &quot;hello, I am a string&quot; starts with &quot;hello, I&quot;.</td>
</tr>
<tr>
<td></td>
<td>NEGATIVE: &quot;hello, I am a string&quot; does not start with &quot;help&quot;.</td>
</tr>
</tbody>
</table>
Operands (Facet Value Extractors)

Facet Value Extractors are utilized by the CylanceOPTICS Context Analysis Engine (CAE) to identify an individual property (Facet) of a single Artifact that was associated with an Event that was observed by CylanceOPTICS. While Facet Value Extractors are narrowly scoped by themselves, they can be strung together in a logical way to analyze complex behaviors that are occurring on a device and trigger a Detection Event in CylanceOPTICS.

<table>
<thead>
<tr>
<th>Extractor Name</th>
<th>Description</th>
<th>Supported Facets</th>
</tr>
</thead>
<tbody>
<tr>
<td>InstigatingProcess</td>
<td>Extracts a facet from the instigating process of an event.</td>
<td>Name (as String)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CommandLine (as String)</td>
</tr>
<tr>
<td>InstigatingProcessImageFile</td>
<td>Extracts a facet from the image file associated with the instigating process of an event.</td>
<td>Path (as String)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Size (as Integer)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Md5Hash (as String)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sha256Hash (as String)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IsHidden (as Boolean)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IsReadOnly (as Boolean)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Directory (as String)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SuspectedFileType (as String)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SignatureStatus (as String)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IsSelfSigned (as Boolean)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LeafDNSString (as String)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LeafThumbprint (as String)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LeafSignatureAlgorithm (as String)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LeafCN (as String)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LeafDN (as String)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LeafOU (as String)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LeafL (as String)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LeafC (as String)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IssuerDNString (as String)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IssuerThumbprint (as String)</td>
</tr>
<tr>
<td>Extractor Name</td>
<td>Description</td>
<td>Supported Facets</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| InstigatingProcessOwner | Extracts a facet from the owner associated with the instigating process of an event. This is commonly used to inspect the user who owns the running process. | Name (as String)  
Domain (as String) |
<p>| TargetFile              | Extracts a facet from the file upon which an event occurred. This is commonly used to inspect various attributes of the file that is being acted upon such as its name, path, hash, or signature status. | See InstigatingProcessImageFile. |
| TargetFileOwner         | Extracts a facet from the owner associated with the file upon which an event occurred. This is commonly used to inspect the user who owns the file being acted upon. | See InstigatingProcessOwner. |
| TargetNetworkConnection | Extracts a facet from the network connection upon which an event               | SourceAddress (as IPAddress)                                                     |</p>
<table>
<thead>
<tr>
<th>Extractor Name</th>
<th>Description</th>
<th>Supported Facets</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetProcess</td>
<td>Extracts a facet from the process upon which an event occurred.</td>
<td>See InstigatingProcess.</td>
</tr>
<tr>
<td></td>
<td>This is commonly used to inspect the name or command line arguments of a</td>
<td></td>
</tr>
<tr>
<td></td>
<td>process being acted upon (like process being started or terminated).</td>
<td></td>
</tr>
<tr>
<td>TargetProcessImageFile</td>
<td>Extracts a facet from the image file associated with a process upon which</td>
<td>See InstigatingProcessImageFile.</td>
</tr>
<tr>
<td></td>
<td>an event occurred.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This is commonly used to inspect various attributes of the image file used</td>
<td></td>
</tr>
<tr>
<td></td>
<td>to launch a process such as its name, path, hash, or signature status.</td>
<td></td>
</tr>
<tr>
<td>TargetProcessOwner</td>
<td>Extracts a facet from the owner associated with a process upon which an</td>
<td>See InstigatingProcessOwner.</td>
</tr>
<tr>
<td></td>
<td>event occurred.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This is commonly used to inspect the user who owns the process being acted</td>
<td></td>
</tr>
<tr>
<td></td>
<td>upon.</td>
<td></td>
</tr>
<tr>
<td>TargetRegistryKey</td>
<td>Extracts a facet from the registry key upon which an event occurred.</td>
<td>Path (as String)</td>
</tr>
<tr>
<td></td>
<td>This is commonly used to inspect the registry key or value that is being</td>
<td>Value (as String)</td>
</tr>
<tr>
<td></td>
<td>acted upon.</td>
<td></td>
</tr>
</tbody>
</table>

### Path Value Extractors

<table>
<thead>
<tr>
<th>Extractor Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnvVar</td>
<td>Extracts an environment variable from the Operating System.</td>
</tr>
<tr>
<td>LiteralWithEnvVar</td>
<td>Expands a path that contains an environment variable.</td>
</tr>
<tr>
<td>Literal</td>
<td>Represents a literal value. This is the most common extractor and operand.</td>
</tr>
</tbody>
</table>
Actions

The Actions field allows users to define a list of Artifacts that can allow CylanceOPTICS to enact automated Response Actions against. The Artifacts of Interest that are able to be defined here follow the same syntax as the Operands defined in the previous section. It should also be noted that any Artifact associated with an Event or set of Events that satisfy a State can be marked as an Artifact of Interest. AOI do not need to be defined as an Operand to be considered an AOI.

In the case that a Filter is applied to a State, users should be aware that some AOI will not be available to take automatic response actions against. For example, if a File Create Filter is applied to a State, users will implicitly have File and Process related AOI available but would not have Registry or Network related AOI. In the event that an irrelevant AOI is provided in a State, the CylanceOPTICS Agent will gracefully handle its exclusion. The table below outlines the applicable Filter to AOI relationships.

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>Type</th>
<th>Applicable AOI</th>
</tr>
</thead>
</table>
| File     |             | Create | InstigatingProcess
<p>|          |             |        | InstigatingProcessImageFile                         |
|          |             |        | InstigatingProcessOwner                             |
|          |             |        | TargetFile                                         |
|          |             |        | TargetFileOwner                                    |
| File     |             | Delete | InstigatingProcess                                |
|          |             |        | InstigatingProcessImageFile                         |
|          |             |        | InstigatingProcessOwner                             |
|          |             |        | TargetFile                                         |
|          |             |        | TargetFileOwner                                    |
| File     |             | Rename | InstigatingProcess                                |
|          |             |        | InstigatingProcessImageFile                         |
|          |             |        | InstigatingProcessOwner                             |
|          |             |        | TargetFile                                         |
|          |             |        | TargetFileOwner                                    |
| File     |             | Write  | InstigatingProcess                                |
|          |             |        | InstigatingProcessImageFile                         |
|          |             |        | InstigatingProcessOwner                             |
|          |             |        | TargetFile                                         |
|          |             |        | TargetFileOwner                                    |</p>
<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>Type</th>
<th>Applicable AOI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>UDP</td>
<td>Connect</td>
<td>InstigatingProcess, InstigatingProcessImageFile, InstigatingProcessOwner, TargetNetworkConnection</td>
</tr>
<tr>
<td>Process</td>
<td>CylancePROTECT</td>
<td>AbnormalExit</td>
<td>TargetProcess, TargetProcessImageFile, TargetProcessOwner</td>
</tr>
<tr>
<td>Registry</td>
<td></td>
<td>PersistencePoint: KeyCreating</td>
<td>InstigatingProcess, InstigatingProcessImageFile</td>
</tr>
<tr>
<td>Category</td>
<td>Subcategory</td>
<td>Type</td>
<td>Applicable AOI</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>--------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessOwner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetRegistryKey</td>
</tr>
<tr>
<td>Registry</td>
<td>PersistencePoint:</td>
<td>KeyCreated</td>
<td>InstigatingProcess</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessImageFile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessOwner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetRegistryKey</td>
</tr>
<tr>
<td>Registry</td>
<td>PersistencePoint:</td>
<td>KeyDeleting</td>
<td>InstigatingProcess</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessImageFile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessOwner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetRegistryKey</td>
</tr>
<tr>
<td>Registry</td>
<td>PersistencePoint:</td>
<td>KeyDeleted</td>
<td>InstigatingProcess</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessImageFile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessOwner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetRegistryKey</td>
</tr>
<tr>
<td>Registry</td>
<td>PersistencePoint:</td>
<td>KeyRenaming</td>
<td>InstigatingProcess</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessImageFile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessOwner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetRegistryKey</td>
</tr>
<tr>
<td>Registry</td>
<td>PersistencePoint:</td>
<td>KeyRenamed</td>
<td>InstigatingProcess</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessImageFile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessOwner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetRegistryKey</td>
</tr>
<tr>
<td>Registry</td>
<td>PersistencePoint:</td>
<td>ValueChanging</td>
<td>InstigatingProcess</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessImageFile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessOwner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetRegistryKey</td>
</tr>
<tr>
<td>Registry</td>
<td>PersistencePoint:</td>
<td>ValueChanged</td>
<td>InstigatingProcess</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessImageFile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessOwner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetRegistryKey</td>
</tr>
<tr>
<td>Registry</td>
<td>PersistencePoint:</td>
<td>ValueDeleting</td>
<td>InstigatingProcess</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessImageFile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessOwner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetRegistryKey</td>
</tr>
<tr>
<td>Category</td>
<td>Subcategory</td>
<td>Type</td>
<td>Applicable AOI</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>-----------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetRegistryKey</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PersistencePoint:</td>
<td>InstigatingProcess</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ValueDeleted</td>
<td>InstigatingProcessImageFile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessOwner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetRegistryKey</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Create</td>
<td>InstigatingProcess</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessImageFile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessOwner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetProcess</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetProcessImageFile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetProcessOwner</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Inject</td>
<td>InstigatingProcess</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessImageFile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>InstigatingProcessOwner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetProcess</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetProcessImageFile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TargetProcessOwner</td>
</tr>
</tbody>
</table>

**Example of Actions:**

```json
"Actions": [ 
  {   "Type": "AOI", 
      "ItemName": "InstigatingProcess", 
      "Position": "PostActivation" 
  },  
  {   "Type": "AOI", 
      "ItemName": "TargetProcess", 
      "Position": "PostActivation" 
  },  
  {   "Type": "AOI", 
      "ItemName": "InstigatingProcessOwner", 
      "Position": "PostActivation" 
  } 
],
```
Filters

Filters allow the scope of a State to be narrowed or expanded to account for a smaller or larger number of events to analyze. Event Filters utilize the same Event Categories, Subcategories, and Types that are outlined in the CylanceOPTICS Sensed Events and Artifacts.

Example 1:

A user wanting to limit inspected Events to Process Start events could structure their Filter section to look like the following:

```
"Filters": [
  {
    "Type": "Event",
    "Data": {
      "Category": "Process",
      "SubCategory": "",
      "Type": "Start"
    }
  }
]
```

Example 2:

Whereas a user wanting to inspect all types of File events (Create, Write, Delete) could structure their Filter section to look like the following (note the wildcard in the Type field):

```
"Filters": [
  {
    "Type": "Event",
    "Data": {
      "Category": "File",
      "SubCategory": "",
      "Type": "*"
    }
  }
]
```

Sensed Events, Artifacts, and Facets

Events, Artifacts, and Facets are three primary data structures that CylanceOPTICS relies on to analyze, record, and investigate activities that occur on devices. A majority of CylanceOPTICS features rely on these data structures - InstaQuery, Focus View, and Context Analysis Engine.

- InstaQuery: Allows a user to search devices for Indicators of Compromise (IOC’s) by querying for specific Artifacts and Facets, like a File Path.
- **Focus View:** Allows a user to view the chain of Events leading up to a piece of malware being introduced to a device by showing a series of Events and each associated Artifact and Facet that make up the Event.

- **Context Analysis Engine (CAE):** Allows CylanceOPTICS to inspect any number of Artifacts and Facets for every Event that occurs on a device, in near-real time.

The following information is an introduction and reference for understanding how CylanceOPTICS interprets and interacts with activities that are occurring on a device to better allow users to utilize Focus Views, InstaQueries, and the Context Analysis Engine.

## Events

Events are defined as the various components that lead to an observable change or action on a device. Events will always consist of two primary Artifacts: the Instigating Artifact that initiates an action, and the Target Artifact that is being acted upon. Each of these Artifacts may consist of Secondary Artifacts (explained below).

CylanceOPTICS is able to sense five main categories of Events that occur on a device:

- File
- Network
- Process
- Registry
- Thread

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>Type</th>
<th>Description</th>
<th>Artifacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>Create</td>
<td>Create</td>
<td>A process has created a file on the device.</td>
<td>Instigating Process, Target File</td>
</tr>
<tr>
<td>File</td>
<td>Delete</td>
<td>Delete</td>
<td>A process has deleted a file off of the device.</td>
<td>Instigating Process, Target File</td>
</tr>
<tr>
<td>File</td>
<td>Write</td>
<td>Write</td>
<td>A process has written to a file.</td>
<td>Instigating Process, Target File</td>
</tr>
<tr>
<td>File</td>
<td>Rename</td>
<td>Rename</td>
<td>A process has renamed a file.</td>
<td>Instigating Process, Target File</td>
</tr>
<tr>
<td>Network</td>
<td>IPv4</td>
<td>Connect</td>
<td>A process has</td>
<td>Instigating</td>
</tr>
<tr>
<td>Category</td>
<td>Subcategory</td>
<td>Type</td>
<td>Description</td>
<td>Artifacts</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>------</td>
<td>-------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Network</td>
<td>IPv4</td>
<td>Connect</td>
<td>established an IPv4 network connection.</td>
<td>Process - Target Network Connection</td>
</tr>
<tr>
<td>Network</td>
<td>IPv6</td>
<td>Connect</td>
<td>A process has established an IPv6 network connection.</td>
<td>Instigating Process - Target Network Connection</td>
</tr>
<tr>
<td>Network</td>
<td>TCP</td>
<td>Connect</td>
<td>A process has established a TCP network connection.</td>
<td>Instigating Process - Target Network Connection</td>
</tr>
<tr>
<td>Network</td>
<td>UDP</td>
<td>Connect</td>
<td>A process has established a UDP network connection.</td>
<td>Instigating Process - Target Network Connection</td>
</tr>
<tr>
<td>Process</td>
<td>Exit</td>
<td>Exit</td>
<td>A process has exited. On Windows, this is observed as the tear down of a process object within the kernel. On Windows, this can typically be accomplished by a call to kernel32!ExitProcess. Currently, CylanceOPTICS does not distinguish between abnormal process exits (crashes or exits under suspicious conditions), normal process exits, and exits under any other conditions. There are many ways in which to cause process exists in Windows for surreptitious reasons.</td>
<td>Instigating Process - Target Process</td>
</tr>
<tr>
<td>Category</td>
<td>Subcategory</td>
<td>Type</td>
<td>Description</td>
<td>Artifacts</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Process</td>
<td></td>
<td>Start</td>
<td>A new process has been created on the device. On Windows, this is observed as the creation of a new process object within the kernel, typically the result of a call to kernel32!CreateProcessW.</td>
<td>Instigating Process, Target Process</td>
</tr>
<tr>
<td>Process</td>
<td>CylancePROTECT</td>
<td>AbnormalExit</td>
<td>The CylancePROTECT service has exited abnormally.</td>
<td>Target Process</td>
</tr>
<tr>
<td>Registry</td>
<td>PersistencePoint: KeyCreating</td>
<td></td>
<td>A process is attempting to create a registry key.</td>
<td>Instigating Process, Target Registry Key</td>
</tr>
<tr>
<td>Registry</td>
<td>PersistencePoint: KeyCreated</td>
<td></td>
<td>A process successfully created a registry key.</td>
<td>Instigating Process, Target Registry Key</td>
</tr>
<tr>
<td>Registry</td>
<td>PersistencePoint: KeyDeleting</td>
<td></td>
<td>A process is attempting to delete a registry key.</td>
<td>Instigating Process, Target Registry Key</td>
</tr>
<tr>
<td>Registry</td>
<td>PersistencePoint: KeyDeleted</td>
<td></td>
<td>A process has successfully deleted a registry key.</td>
<td>Instigating Process, Target Registry Key</td>
</tr>
<tr>
<td>Registry</td>
<td>PersistencePoint: KeyRenaming</td>
<td></td>
<td>A process is attempting to rename a registry key.</td>
<td>Instigating Process, Target Registry Key</td>
</tr>
<tr>
<td>Registry</td>
<td>PersistencePoint: KeyRenamed</td>
<td></td>
<td>A process has successfully renamed</td>
<td>Instigating Process</td>
</tr>
<tr>
<td>Category</td>
<td>Subcategory</td>
<td>Type</td>
<td>Description</td>
<td>Artifacts</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
</tbody>
</table>
| Registry | PersistencePoint: ValueChanging | A process is attempting to change the value of a registry key. | - Instigating Process  
- Target Registry Key |
| Registry | PersistencePoint: ValueChanged | A process has successfully changed the value of a registry key.                         | - Instigating Process  
- Target Registry Key |
| Registry | PersistencePoint: ValueDeleting | A process is attempting to delete the value of a registry key.                         | - Instigating Process  
- Target Registry Key |
| Registry | PersistencePoint: ValueDeleted | A process has successfully deleted the value of a registry key.                         | - Instigating Process  
- Target Registry Key |
| Thread   | Create      | A thread has been created by a process within itself. On Windows, this means that a process has either called kernel32!CreateProcess and the first thread in the new process has been created or called ntdll!NtCreateThread to create a new thread within the same process. | - Instigating Process  
- Target Process  
- Target Thread |
| Thread   | Inject      | A thread has been created by one process into another.                               | - Instigating Process  
- Target Process |
<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>Type</th>
<th>Description</th>
<th>Artifacts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Threads cannot be created in one process and then moved to another, so this event signals the start of a new thread at its point of first execution. On Windows, this means that a process has called ntdll!CreateThread, specifying a different process for the target of the new thread object.</td>
<td>■ Target Thread</td>
</tr>
</tbody>
</table>

**Artifacts and Facets**

Artifacts are complex pieces of information and can be utilized within CylanceOPTICS. The Context Analysis Engine utilizes Artifacts of Interest (AOI) as specifically tagged Artifacts that Automated Response Actions can be applied to for automatic incident response and incident remediation while InstaQuery utilizes Artifacts as the foundation of a query. CylanceOPTICS utilizes six Artifacts:

- File
- Network Connections
- Processes
- Registry Keys
- Threads
- Users

Facets are attributes of Artifacts that can be used to identify traits or qualities of an Artifact that is associated with an Event. Facets can be logically combined during analysis for further conviction of maliciousness or suspiciousness. For example, a file named "explorer.exe" may not be inherently suspicious; however, if a file named "explorer.exe" is not signed by Microsoft, and resizes in a temporary directory, it may be deemed as suspicious in some environments. The table below outlines the Artifacts and their associated Facets that are currently supported by CylanceOPTICS.
<table>
<thead>
<tr>
<th>Artifact</th>
<th>Facets</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>Path, File Size, Creation Time, Last Modified Time, Owner (see User artifact), MD5 Hash, SHA256 Hash, Is Read Only, Is Hidden, Is On Removable Drive, Suspected File Type</td>
</tr>
<tr>
<td>Network</td>
<td>Source IP Address, Source Port, Destination IP Address, Destination Port</td>
</tr>
<tr>
<td>Process</td>
<td>Process ID, Process Name, Primary Image File (see File artifact), Owner (see Owner artifact), Command Line, Description, Start Time, End Time, Parent Process (see Process artifact), Session ID (Windows only)</td>
</tr>
<tr>
<td>Registry</td>
<td>Key Path, Value Name, Is Persistence Point?</td>
</tr>
<tr>
<td>Thread</td>
<td>Start Address, Module (see File artifact), Resolved Function</td>
</tr>
<tr>
<td>User</td>
<td>Name, Domain</td>
</tr>
</tbody>
</table>