

SECUREWORKS TAEGIS XDR INTEGRATION WITH SECURITY VALIDATION

This integration lets you collect alerts generated by Secureworks Taegis XDR to test the efficacy and configuration of the security control using Security Validation jobs.

API Calls

API	Usage
<code>/auth/api/v2/auth/token</code>	Retrieve token from Secureworks Taegis XDR
<code>/graphql</code>	Query with GraphQL alerts and assets detected on endpoints with the Secureworks agent installed

Supported Versions

Secureworks Taegis XDR v1

Before You Begin

To configure this integration you will need,

- The hostname of your XDR API instance (can be found on **Regions**)
- A Secureworks Client ID
- A Secureworks Client Secret

Regions

- US1: `https://api.ctpx.secureworks.com`
- US2: `https://api.delta.taegis.secureworks.com`
- US3: `https://api.foxtrot.taegis.secureworks.com`
- EU: `https://api.echo.taegis.secureworks.com`

Generate a Secureworks Taegis XDR API Key

For more information, see the [Secureworks API documentation \(https://docs.ctpx.secureworks.com/apis/api_authenticate/\)](https://docs.ctpx.secureworks.com/apis/api_authenticate/).

1. Log in to XDR in Chrome.
2. Open the **Chrome Developer Tools**.
3. Go to the **Console** tab and enter the following:

```
copy(localStorage.access_token)
```

4. An `access_token` is copied to your clipboard. Note that the token is not displayed in the **Chrome Developer Tools Console**; it is only copied to your clipboard. The command returns `undefined`.
5. In a command line terminal, run the following commands to create your client credentials, depending on which platform you're using:
 - **Credentials on Linux**

```
export ACCESS_TOKEN="YOUR_ACCESS_TOKEN"
export TENANT_ID="YOUR_TENANT_ID"
curl -g \
-H "Authorization: Bearer $ACCESS_TOKEN" \
-H "X-Tenant-Context: $TENANT_ID" \
-H "Content-type: application/json" \
-X POST \
-d '{"query": "mutation createClient($name: String!, $roles: [ID!]) { createClient(name: $name, roles: $roles)
{ client { id name client_id roles role_assignments { id tenant_id role_id role_name expires_at }
tenant_id created_at updated_at created_by updated_by environment } client_secret } }", "variables":
{"name": "YOUR_UNIQUE_APPLICATION_NAME"}' \
https://api.ctpx.secureworks.com/graphql
```

o Credentials on Windows

```
set ACCESS_TOKEN=YOUR_ACCESS_TOKEN
set TENANT_ID=YOUR_TENANT_ID
curl -H "Authorization: Bearer %ACCESS_TOKEN%" -H "X-Tenant-Context: %TENANT_ID%"
-H "Content-type: application/json" https://api.ctpx.secureworks.com/graphql -d
"{\"query\": \"mutation createClient($name: String!, $roles: [ID!]) { createClient(name: $name, roles: $roles)
{ client { id name client_id roles role_assignments { id tenant_id role_id role_name expires_at }
tenant_id created_at updated_at created_by updated_by environment } client_secret } }\",
\"variables\": {\"name\": \"YOUR_UNIQUE_APPLICATION_NAME\"}}"
```

- **YOUR_ACCESS_TOKEN**: Paste the `access_token` from your clipboard.
- **YOUR_TENANT_ID**: Replace with the tenant ID value for your environment.
- **YOUR_UNIQUE_APPLICATION_NAME**: Enter a unique name to identify your application.

6. Your new client credentials are returned. Save the `client_id` and `client_secret` values from this response — your application needs these to sign in to the XDR system. These are equivalent to a username and password for a user, so please protect them as you would any other login credentials.

Configure the Integration in Security Validation

1. Go to **Settings > Integrations**.
2. From the Integrations table, click **Add Integration > Secureworks Taegis XDR**.



You can add this as either a Direct or Remote Integration.

3. Enter a meaningful **Integration Name**.
4. Optional: From the **Proxy** drop-down, choose a proxy profile if one is available. If one isn't available and all outbound connections go through a proxy, first, set up a **Proxy Rule** (<https://docs.mandiant.com/home/msv-proxy-rules>).
5. Change the **HttpProtocols** to use for requests. (**Https** or **Http**.)
6. For the **Host**, change the value if needed. The default is **httpbin.org**.
7. Enter a **Port** value. The default is **443**.
8. Enter the **Client Id** and **Client Secret** values that you generated.
9. Enter the **Tenant Id** value for your environment.
10. Optional: Check **Verify Ssl** if you want this verification done for requests to an upstream server.
11. Optional: Change the **Timeout** value if you want a different frequency of requests to an upstream server. The default is **30** (seconds).
12. Add values for **Logsets**. These typically correspond to an event type. For more information, see the **Rapid7**

InsightIDR logset documentation (<https://docs.rapid7.com/insightidr/log-sets-you-can-search/>).

13. Optional: Change **Queries**, if needed. A default value is provided.
14. Optional: Modify the **Field Map** values, as necessary.



- Each field map box can hold a JSON-formatted comma-separated list of columns returned by the API to be considered for each field when translating into the normalized event object format. Example: description could be configured to be 'msg_s' or 'SyslogMessage' in some environments. The field map tries both if set to: ['msg_s','SyslogMessage'] and whichever matches first is the column that is used.
- When configuring an integration in Security Validation, you can assign additional host values in the Field Map settings. If none of the assigned fields return a valid host name, Network Actions may miss matched events from the third-party technology. Additional hosts values helps ensure the likelihood of a match between the two environments.

15. Optional: Modify the **Page Size** value for requests to the upstream server. The default is **500**.
16. Optional: Expand **Advanced options** and update the information as necessary.
 - a. Update **Query Time** and **Delay Time**.



The **Query time** is the amount of time (minutes) before and after the query runs that the platform looks for events, while the **Delay time** is the amount of time (minutes) that the platform waits to run the first query after a Job Action starts. For example, you configure your integration with the following values: **Query time** = 5, **Query interval** = 30 seconds, and **Delay time** = 0. When a Job Actions starts at 12:00:00, the first time the query runs, the platform looks for events from 11:55:00 to 12:00:00. Then 30 seconds later, it looks for events from 11:55:30 to 12:00:30. This interval continues, with the last query looking from 12:00:00 to 12:05:00. If you instead configured the **Delay time** to equal 10, it would run the same query, but it wouldn't start that query until 12:10:00.



If your monitors are set to run more frequently than the query time, this configuration impacts the pass/fail results for AEDA monitors.

- b. Update **Query Interval** (seconds).
- c. Select **Correlation Query Enabled** and fill in the **Correlation Query**.
- d. Modify the **Correlation Query Interval**, if necessary (minutes).
- e. Select **Discover network devices automatically**, the default and recommended option.




If unselected, reported events won't include product information for any matching network security technology.

- f. Select **Save Suspicious Events**.
- g. Modify the **Event Time Adjustment** (seconds). The default is **0**.
- h. Modify the **Limit** value if you need to prevent a flood of results. This value is set to **10000** by default. This limit applies to both events and alerts individually, so if you set it to **10**, you can still see a maximum of 10 events and 10 alerts.

17. Click **Save**.

Verify connectivity

1. Go to **Settings > Integrations**.
2. From the Direct Integrations table, click  **> Test** to verify that:

- The Director can communicate with the integration host on the port and protocol specified.
- The integration credentials are valid and working.

For more information on setting up queries, see **Manage Integrations** (<https://docs.mandiant.com/home/msv-managing-integrations>).