Mandiant

SNOWFLAKE INTEGRATION WITH SECURITY VALIDATION

This integration provides the following benefits:

- Validates that security tools are writing log events to Snowflake to ensure compliance with security policies and regulations
- Collects events generated by security tools that write to Snowflake to test the efficacy and configuration of security controls using Security Validation jobs

API Calls

API	Usage
`/api/v2/statements`	Retrieve Events from any table in Snowflake

Supported Versions

Snowflake (Cloud)

Before You Begin

To configure this integration, you need:

- The hostname of your Snowflake Cloud instance
- Your account_identifier for your account. See the **Account Identifier documentation (https://docs.snowflake.com/en/user-guide/admin-account-identifier.html)** for more information.
- A valid username and password for an account with access to the resources you wish to query
- The Warehouse to use when making a query
- The Database to use when making a query
- The role the use for the given username with proper permissions for the query being executed.

Snowflake Documentation Links

- Snowflake REST API `/statements` : **https://docs.snowflake.com/en/developer-guide/sql-api/reference#label-sql-api-reference-post-statements-request-body**
- Snowflake OAuth Athentication: **https://docs.snowflake.com/en/developer-guide/sql-api/authenticating**

Configure Security Validation

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

   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. Optional: Change the **HttpProtocols** value to determine what protocol is used for requests (**Https** or **Http**).
6. Enter the **Host** for the Snowflake instance. The default is **app.snowflake.com**.
7. Enter a **Port** value. The default is **443.**
8. Enter the **Account Identifier**.
9. Enter the **Username** and **Password** for the account with permission to use API endpoints.
10. Optional: Enter the **Role** that the configured user should use when running queries. The default is **ACCOUNTADMIN**.
11. Optional: Enter the **Warehouse** to execute queries within. If left blank, the default is the user

DEFAULT_NAMESPACE.

12. Optional: Enter the **Database** to execute queries on. If left blank, the default is the user **DEFAULT_NAMESPACE**.
13. Optional: Check **Verify Ssl** if you want this verification done for requests to an upstream server.
14. Optional: Change the **Timeout** value if you want a different frequency of requests to an upstream server. The default is **30** (seconds).
15. Optional: Modify **Queries**, as needed. Defaults are provided for **IP Query**.
16. 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.

17. Optional: Modify the **Page Size** to change the request for the upstream server. The default is **500**.
18. 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.

19. 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)**.

Troubleshooting

Error message: `unhandled errors in a TaskGroup (1 sub-exception), An unknown error occurred`

- This indicates a snowflake error in the query provided. Recheck the following:
    - user credentials (user, pass, role, warehouse, database, schema, table)
    - user permissions to the tables (etc) in snowflake
    - query for syntax issues

Example Data and Query

The default query was created base on this sample table data:

```
create table "events" (
   "uid" TEXT,
   "src_ip" TEXT
   "src_port" NUMBER
   "dest_ip" TEXT
   "dest_port" NUMBER
   "start_time" DATETIME
   "sid" TEXT
   "url" TEXT
   "email_sender" TEXT
   "email_recipient" TEXT
   "email_subject" TEXT
   "description" TEXT
   "host" TEXT
   "computer" TEXT
   "user" TEXT
   "filehashes" TEXT
   -- , <col1_name> <col1_type>
   -- supported types: https://docs.snowflake.com/en/sql-reference/intro-summary-data-types.html
   )
   -- comment = '<comment>';
```