EasyDMARC & Microsoft Sentinel Integration: Customer Setup Guide

This guide will walk you through the necessary steps to configure your Microsoft Azure and Sentinel environment to receive audit logs and alerts from your EasyDMARC service. This integration will use the Azure Monitor Logs Ingestion API, allowing EasyDMARC to securely send data directly to your Sentinel workspace.

Before you begin, please ensure you have:

- An active Azure subscription.
- A Log Analytics workspace with Microsoft Sentinel enabled.
- Permissions to:
    - Register applications in Microsoft Entra ID.
    - Create and manage tables in your Log Analytics workspace.
    - Create and manage Data Collection Rules (DCRs) and Data Collection Endpoints (DCEs) in Azure Monitor.
    - Assign roles (RBAC) on Azure resources (specifically DCRs).

Overview of Steps:

1. **Create a Microsoft Entra ID Application Registration:** This app will represent EasyDMARC in your Azure environment for authentication.
2. **Create Custom Log Tables in Log Analytics:** These tables will store your EasyDMARC audit logs and alerts.
3. **Configure Data Collection Rules (DCRs):** These rules will define how data sent by EasyDMARC is processed and routed to your custom tables.
4. **Grant Permissions to DCRs:** Allow the registered application to send data to your DCRs.

5. **Collect Information for EasyDMARC:** Gather all the necessary IDs and endpoints to configure the integration in your EasyDMARC portal.

Step 1: Create a Microsoft Entra ID Application Registration

This application registration will allow EasyDMARC to authenticate securely to your Azure environment to send logs.

1. **Navigate to Azure Portal:**
    - Sign in to the Azure portal.
2. **Go to Microsoft Entra ID:**
    - In the Azure portal search bar, type "Microsoft Entra ID" and select it.
3. **Access App Registrations:**
    - In the Microsoft Entra ID navigation pane, select **App registrations**.
4. **Create New Registration:**
    - Click on **+ New registration**.
    - **Name:** Enter a descriptive name, for example, `EasyDMARC-Sentinel-Integration`.
    - **Supported account types:** Select **"Accounts in this organizational directory only (Your Organization Name only - Single tenant)"**.
    - **Redirect URI (optional):** You can leave this blank for this type of service-to-service integration.
    - Click **Register**.
5. **Record Application Identifiers:**
    - Once the application is registered, you'll be taken to its overview page.
    - Copy and securely store the **Application (client) ID** and the **Directory (tenant) ID**. You will need these later.
6. **Create a Client Secret:**

- In the application's navigation pane (e.g., for `EasyDMARC-Sentinel-Integration`), select **Certificates & secrets**.
- Under the **Client secrets** tab, click **+ New client secret**.
- **Description:** Enter a description, for example, `EasyDMARCSentinelIntegrationSecret`.
- **Expires:** Select an expiry period (e.g., 12 months, 24 months). **Note the expiry date and plan to renew it before it expires to avoid service interruption.**
- Click **Add**.
- **IMPORTANT:** Immediately copy the **Value** of the client secret. This secret value **will not be shown again** after you leave this page. Store it securely (e.g., in a password manager or secure note). This is *not* the "Secret ID".

Step 2: Create Custom Log Tables in Your Log Analytics Workspace

You will create two separate custom tables: one for your EasyDMARC audit logs and one for EasyDMARC alerts. To help Azure define the structure (schema) for these tables correctly, you will use sample JSON files provided by EasyDMARC.

1. **Navigate to Your Log Analytics Workspace:**
   - In the Azure portal search bar, type "Log Analytics workspaces" and select your workspace that is connected to Microsoft Sentinel.
2. **Go to Tables:**
   - In the workspace navigation pane, under "Settings," select **Tables**.
3. **Create Custom Table for Audit Logs:**
   - Click **Create** and select **New custom log (DCR-based)**.

- **Table Name:** Enter `EasyDMARCAuditLogs` (the portal will automatically append `_CL`, making it `EasyDMARCAuditLogs_CL`).
- Click **Next: Data Collection Rule >**.
- **Upload sample log file:**
  - **Download the EasyDMARC Audit Log sample JSON file from:** Sample JSON *(Right-click and "Save Link As..." or click to download)*
  - Once downloaded, click **Browse for files** on the Azure portal page and upload the `audit_log.json` (or the name you give it, e.g., `easydmarc_audit_sample.json`) file you just downloaded. This sample helps Azure infer the initial table schema.
- Review the inferred schema to ensure it matches the fields that will be sent by EasyDMARC (as detailed in further EasyDMARC integration documentation if needed). Confirm the table creation.

4. **Create Custom Table for Alerts:**
   - Repeat the process above: Click **Create** -> **New custom log (DCR-based)**.
   - **Table Name:** Enter `EasyDMARCAlerts` (will become `EasyDMARCAlerts_CL`).
   - **Upload sample log file:**
     - **Download the EasyDMARC Alerts sample JSON file from:** Sample JSON *(Right-click and "Save Link As..." or click to download)*
     - Once downloaded, click **Browse for files** and upload the `alerting.json` (or the name you give it, e.g., `easydmarc_alert_sample.json`) file.
   - Review the inferred schema and confirm the table creation.

Note from EasyDMARC: These sample JSON files are crucial for ensuring your custom tables are created with the correct fields and data types to receive our audit logs and alerts. The specific field definitions and detailed schemas are also available in our main integration documentation for your reference during DCR configuration in the next step.

Step 3: Configure Data Collection Rules (DCRs)

DCRs define how Azure Monitor collects and processes incoming data before it's sent to your Log Analytics tables. A basic DCR might have been created during table creation in Step 2. You may need to review and refine it, or create new ones specifically for this API ingestion.

1. **Navigate to Data Collection Rules:**
    - In the Azure portal search bar, type "Monitor" and select it.
    - In the Monitor navigation pane, under "Settings," select **Data Collection Rules**.
2. **Create or Identify DCRs:**
    - You will need two DCRs: one for `EasyDMARCAuditLogs_CL` and one for `EasyDMARCAlerts_CL`. If basic DCRs were created during table creation, locate them. Otherwise, click **+ Create**.
3. **Configure Each DCR:**
    - **Basics Tab:**
        - **Rule name:** (e.g., `DCR-EasyDMARC-AuditLogs` or `DCR-EasyDMARC-Alerts`).

        - **Subscription, Resource Group, Region:** Choose appropriate values. The region should generally match your Log Analytics workspace.
        - **Platform Type:** Select `Windows` or `Linux` (this setting is less critical for direct API ingestion but still required by the portal).
        - **Data Collection Endpoint (DCE):**
            - If you don't have one, you may need to create a Data Collection Endpoint by clicking **Create new**. A DCE provides a unique ingestion endpoint. Ensure it's in the same region as your Log Analytics workspace. Note its **Logs ingestion URI**.
            - Alternatively, DCRs with `kind: "Direct"` (which is what the Logs Ingestion API uses) will have their own ingestion endpoint.
    - **Resources Tab:** This is typically for associating DCRs with VMs if using agents. For API ingestion, this might not be directly configured here unless you are associating the DCR with a DCE that serves a broader purpose.
    - **Collect and deliver Tab (Data Sources & Destinations):** This is where you define the core logic. You might need to edit the DCR in its JSON view for full control over `streamDeclarations` and `transformKql`.
        - **Go to the JSON View of the DCR:** Once the DCR is created (even a basic one), find it in the list, select it, and go to "JSON View".
        - **`streamDeclarations`:**
            - This section defines the schema of the data EasyDMARC will send. EasyDMARC **must provide you with the exact stream names and JSON schemas** for both audit logs and alerts.

    - Example stream name for audit logs: `Custom-EasyDMARCAuditStream`
    - Example stream name for alerts: `Custom-EasyDMARCAlertStream`
    - You'll add/modify this in the DCR JSON. Each stream will have a list of columns with their names and types (e.g., `TimeGenerated_d: "datetime"`, `User_s: "string"`).
- **`destinations`:**
    - This will point to your Log Analytics workspace. This is usually configured correctly if the DCR was linked during table creation.
- **`dataFlows`:**
    - This section connects the input `streams` (from `streamDeclarations`) to the `destinations` (your custom tables) and applies a KQL transformation.
    - **`transformKql`**: EasyDMARC **must provide you with recommended KQL queries** to map the fields from their JSON structure (defined in your `streamDeclarations`) to the columns in your `EasyDMARCAuditLogs_CL` and `EasyDMARCAlerts_CL` tables. A key transformation is ensuring the `TimeGenerated` column in your table is populated correctly (e.g., `source | project TimeGenerated = todatetime(TimeGenerated_d_datetime), UserPrincipalName = User_s, /* other mappings */`).

        - **`outputStream`**: This will be the name of your custom table, prefixed with `Custom-` (e.g., `Custom-EasyDMARCAuditLogs_CL`).
4. **Record DCR Information:** For *each* DCR:
    - **DCR Immutable ID:** Find this on the "JSON View" or "Overview" page of the DCR.
    - **Logs Ingestion Endpoint URI:** Find this in the "JSON View" of the DCR. It will be under `properties.logsIngestion.endpoint`. (Alternatively, if you are using a dedicated DCE, use its `logsIngestion` URI).
    - **Stream Name:** The exact name you defined in the `streamDeclarations` section of the DCR JSON for audit logs and alerts respectively (e.g., `Custom-EasyDMARCAuditStream`, `Custom-EasyDMARCAlertStream`).

Step 4: Grant Permissions to Data Collection Rules (DCRs)

The Entra ID application you created in Step 1 needs permission to send data to these DCRs.

1. **Navigate to Each DCR:** Go to "Monitor" -> "Data Collection Rules," and select one of the DCRs you configured (e.g., `DCR-EasyDMARC-AuditLogs`).
2. **Access Control (IAM):** In the DCR's navigation pane, select **Access control (IAM)**.
3. **Add Role Assignment:**
    - Click **+ Add** -> **Add role assignment**.
    - **Role:** Search for and select **Monitoring Metrics Publisher**.

- **Assign access to:** Select **User, group, or service principal**.
- **Members:** Click **+ Select members**. Search for the name of the application you created in Step 1 (e.g., `EasyDMARC-Sentinel-Integration`). Select it and click **Select**.
- Click **Review + assign**, and then **Review + assign** again.

4. **Repeat for the other DCR** (e.g., `DCR-EasyDMARC-Alerts`).

Step 5: Collect and Provide Information to EasyDMARC

Gather all the following information. You will need to enter this into your EasyDMARC platform's Sentinel integration setup page:

1. **Directory (Tenant) ID:** (From Step 1.5)
2. **Application (Client) ID:** (From Step 1.5)
3. **Client Secret Value:** (The secret value you copied in Step 1.6)
4. **For Audit Logs Data:**
   - DCR Immutable ID (for the audit logs DCR - from Step 3.4)
   - Logs Ingestion Endpoint URI (for the audit logs DCR - from Step 3.4)
   - Stream Name (for audit logs, as defined in the DCR, e.g., `Custom-EasyDMARCAuditStream` - from Step 3.4)
5. **For Alerts Data:**
   - DCR Immutable ID (for the alerts DCR - from Step 3.4)
   - Logs Ingestion Endpoint URI (for the alerts DCR - from Step 3.4)
   - Stream Name (for alerts, as defined in the DCR, e.g., `Custom-EasyDMARCAlertStream` - from Step 3.4)

Final Steps:

- Once you have provided this information to EasyDMARC, they will use it to configure the data flow from their platform to your Sentinel workspace.
- Utilize any "Test Connection" feature provided within the EasyDMARC platform to ensure the setup is correct.
- Allow some time for data to start appearing in your `EasyDMARCAuditLogs_CL` and `EasyDMARCAlerts_CL` tables in Log Analytics / Sentinel. You can then start building analytics rules, workbooks, and hunting queries based on this data.

EasyDMARC will provide the specific sample JSON log files for table creation, the exact JSON schema definitions for the `streamDeclarations` in your DCRs, and recommended KQL queries for the `transformKql` section of your DCRs.

Example KQL queries:

Alerts

```
EasydmarcAlerts_CL
| where Severity == "CRITICAL"
```

Audit Logs

```
EasydmarcAuditLogs_CL
| where Action in (
    "DOMAIN ADDED",
    "DOMAIN REMOVED",
    "MANAGED DMARC ACTIVATED",
    "MANAGED DMARC RECORD CHANGED",
    "FAILURE REPORT DISABLED",
```

```
    "FAILURE REPORT ENABLED",
    "USER INVITED AS ADMIN",
    "USER INVITED AS EDITOR",
    "USER INVITED AS VIEWER",
    "USER REVOKED FROM ORGANIZATION",
    "USER PERMISSION CHANGED"
)
```