Measure Killer Security Information and Documentation

Version 2.6.0

This document outlines the security details and data handling practices of Measure Killer.

Data Handling and Privacy Assurance

Measure Killer is meticulously designed with user privacy and data security as paramount concerns. It is important for users to understand the scope and way that Measure Killer interacts with Power BI.

Metadata-Only Interaction

Measure Killer only interacts and utilizes metadata from Power BI Desktop or the Power BI Service. At no point does Measure Killer read, store, or process actual data contained within Power BI reports or models. This strict limitation to metadata ensures that sensitive data remains confidential and secure.

What is considered Metadata?

Metadata is considered the following (examples):

- Names or artifacts (report, semantic models, workspaces, users/email addresses)
- DAX and M expressions including comments
- Names of measures, tables, calculated columns, calculated tables, calculation groups, field parameters etc.
- Uncompressed size of columns
- All definitions (e.g. name of reports, pages, visuals, visual titles, visual size and other metadata)

Local Processing of Metadata

All processing and analysis of metadata conducted by Measure Killer is performed locally on the user's machine through the installed client software. This approach ensures that the operations on metadata do not involve or require any external processing or storage facilities. By doing so, Measure Killer upholds a high standard of data protection and minimizes potential security risks. There is no database that is hosted, managed, or used to process any kind of information, everything is done on the user’s machine and not transmitted anywhere besides when using XMLA or REST API calls with Microsoft directly.

XMLA Endpoint Connection

To retrieve metadata from semantic models in the Power BI Service, XMLA endpoints are used. The authentication happens via the entered Microsoft account (see adomd client for more specifics on this).

API Calls to Microsoft for Metadata Acquisition

The only other external interaction Measure Killer undertakes involves REST API calls to Microsoft’s Power BI Service for the acquisition of metadata. These API calls are made in compliance with all relevant security protocols. Authentication happens via browser interaction (OAUTH2). Once metadata has been acquired, all subsequent analyses and operations are carried out locally. See the list at the bottom of this document for a list of REST API calls done.

Online / Offline Modes

Measure Killer is designed to be able to operate both online and offline:

- In the offline modes, Measure Killer connects to the local instance of Analysis Services to retrieve the metadata it needs (Modes 1 and 2). In these modes the only time there is an interaction with the internet is upon launch to verify valid licensing (see License Verification below for details)

- Online modes (Modes 3, 4 and 5) are triggered when an active internet connection is detected and when the user has a valid license. In these modes there are external connections happening (only REST API calls and XMLA connections to Microsoft as outlined above).

License Verification and Datetime Check

- Measure Killer uses API calls to acquire the current date and time: GET https://worldtimeapi.org/api/timezone/Europe/London or, as a fallback, GET https://timeapi.io/api/Time/current/zone?timeZone=Europe/London to ensure the license has not expired.

Version Check and User Notification

- To inform users about available updates, Measure Killer queries GET https://measurekiller.com/downloads/ to compare the installed version against the most recent version available the website.

- There is one additional GET request to extract text and write it in the main UI window from this URL GET https://en.brunner.bi/post/measure-killer-feedback-1

API Endpoints and Security Measures

- Secure requests (HTTPS) are made to Power BI API endpoints, ensuring data in transit is encrypted.

- Access tokens are securely passed in request headers, with responses parsed as JSON for internal processing.

List of REST API Calls

- Datasets - Get Datasets In Group: GET https://api.powerbi.com/v1.0/myorg/groups/{workspace.id}/datasets

- Datasets - Get Refresh History In Group: GET https://api.powerbi.com/v1.0/myorg/groups/{workspace.id}/datasets/{dataset.id}/refreshes

- Datasets - Get Refresh History: GET https://api.powerbi.com/v1.0/myorg/datasets/{dataset.id}/refreshes

- Datasets - Execute Queries In Group: GET https://api.powerbi.com/v1.0/myorg/groups/{workspace.id}/datasets/{dataset.id}/executeQueries

- Reports - Get Reports In Group: GET https://api.powerbi.com/v1.0/myorg/groups/{workspace.id}/reports

- Reports - Get Datasources In Group: GET https://api.powerbi.com/v1.0/myorg/groups/{workspace.id}/reports/{report.id}/datasources

- Reports - Get Datasources: GET https://api.powerbi.com/v1.0/myorg/reports/{report.id}/datasources

- Reports - Export Report In Group: GET https://api.powerbi.com/v1.0/myorg/groups/{workspace.id}/reports/{report.id}/Export

- Reports - Export Report: GET https://api.powerbi.com/v1.0/myorg/reports/{report.id}/Export

- Groups - Get Groups: GET https://api.powerbi.com/v1.0/myorg/groups/{workspace.id}/users

- Groups - Get Groups: GET https://api.powerbi.com/v1.0/myorg/groups

List of REST API Admin Calls

- Admin - Get Capacities As Admin: GET https://api.powerbi.com/v1.0/myorg/admin/capacities

- Admin - WorkspaceInfo GetScanStatus: GET https://api.powerbi.com/v1.0/myorg/admin/workspaces/scanStatus/{scan.id}

- Admin - WorkspaceInfo GetScanResult: GET https://api.powerbi.com/v1.0/myorg/admin/workspaces/scanResult/{scan.id}

- Admin - WorkspaceInfo PostWorkspaceInfo: POST https://api.powerbi.com/v1.0/myorg/admin/workspaces/getInfo

- Admin - Get Activity Events: POST https://api.powerbi.com/v1.0/myorg/admin/activityevents

- Admin - Groups GetGroupsAsAdmin: GET https://api.powerbi.com/v1.0/myorg/admin/groups

- Admin - Groups AddUserAsAdmin: POST https://api.powerbi.com/v1.0/myorg/admin/groups/{workspace.id}/users

- Admin - Groups DeleteUserAsAdmin: DELETE https://api.powerbi.com/v1.0/myorg/admin/groups/{workspace.id}/users/{user.email}

- Admin - Users GetUserArtifactAccessAsAdmin: GET https:/api.powerbi.com/v1.0/myorg/admin/users/{user.email}/artifactAccess

List of Fabric API Calls

- Items - Get Item Definition: POST https://api.fabric.microsoft.com/v1/workspaces/{workspace.id}/items/{item.id}/getDefinition

- Long Running Operations - Get Operation State: GET https://api.fabric.microsoft.com/v1/operations/{operation.id}

- Long Running Operations - Get Operation Result: GET https://api.fabric.microsoft.com/v1/operations/{operation.id}/result

- Domains - List Domain Workspaces GET https://api.fabric.microsoft.com/v1/admin/domains/{domainId}/workspaces

- Domains - List Domains GET https://api.fabric.microsoft.com/v1/admin/domains

WIX

- Create Contact: POST https://www.wixapis.com/contacts/v4/contacts

Network time Protocol (NTP)

- Measure Killer uses NTP via the ntplib library in python to enforce license expiration