Integration of Intel MQ with a PostgreSQL database.

Intel MQ installation and configuration: Using the correct login credentials from Rads (username: intelapi, password: intelapi), I successfully accessed the IntelMQ user interface. Subsequently, I configured and managed various bots for diverse data outputs. Access the server where IntelMQ is running. This can be done via SSH if you are working on a remote server.

To facilitate data storage and management, I set up a dedicated Ubuntu virtual machine to host the PostgreSQL database. Following the official PostgreSQL documentation, I installed PostgreSQL version 10.

After completing the PostgreSQL installation, I created a new database named " mydtmdb// mydb using the following command:

```
sudo -u postgres createdb threatdata
```

Within the " mydtmdb " database, I created a new table called 'events' to store the data. The table was created using the following SQL statement:

```
CREATE TABLE events (
    id SERIAL PRIMARY KEY,
    event_type TEXT,
    event_data JSONB,
    timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Testing connectivity

Testing the connectivity between the IntelMQ instance and the Ubuntu VM hosting the PostgreSQL database was successful, as confirmed by successful ping responses from both ends.

I inserted some dummy data into the 'events' table using the following INSERT statement:

```
INSERT INTO events (event_type, event_data)
VALUES ('test_event', '{"source": "test", "data": "dummy data"}');
```

After inserting the data, I verified that the data was successfully inserted by running the following query:

```
SELECT * FROM events;
```

In the IntelMQ configuration file located at /etc/intelmq/runtime.yaml, I defined an output section for the PostgreSQL output bot named "postgresql-output". I specified the necessary parameters such as the database connection string, table name, and other relevant settings.

Upon initiating the execution of the configured bots to process the data flow, the expected output was generated, and the data was successfully stored in the PostgreSQL database. However, during the integration process, I encountered the following challenges:

Challenges:

Database Connection String: Initially, there was an issue with the database connection string format, which prevented IntelMQ from connecting to the PostgreSQL database successfully.

User Permissions: Ensuring that the user running IntelMQ had the necessary permissions to access and write to the PostgreSQL database required additional configuration steps.

Troubleshooting Steps:

Database Connection Verification: I verified the database connection string by attempting to connect to the PostgreSQL database using a separate client, such as pgAdmin / psql.

User Permission Review: I reviewed the user permissions and granted the necessary privileges to the user running IntelMQ to access and write to the PostgreSQL database.

Log Examination: I examined the IntelMQ logs for any errors or warnings related to the database connection or writing data to the database.

Findings from the Intel MQ and PostgreSQL Database Integration task:

Successes: Access to Intel MQ Interface: Successfully accessed the Intel MQ user interface using provided credentials.

Bot Configuration: Configured and managed various bots for data outputs within the Intel MQ environment.

PostgreSQL Installation: Installed PostgreSQL on a dedicated Ubuntu virtual machine to serve as the database backend.

Database Setup: Created a new database named " mydtmdb " and initialized a table named 'events' within it for data storage.

Connectivity Establishment: Successfully established ping connectivity between the Intel MQ instance and the dedicated Ubuntu VM.

Data Insertion: Successfully inserted dummy data into the 'events' table for testing purposes.

Bot Integration: Configured the SQL-Output bot through the IntelMQ configuration file, specifying necessary parameters such as the database connection string, table name, and other relevant settings.

Data Output: After resolving the initial challenges, the SQL-Output bot successfully stored the data in the PostgreSQL database.

Challenges:

Database Connection String: Encountered an issue with the database connection string format, preventing IntelMQ from connecting to the PostgreSQL database.

User Permissions: Ensuring that the user running IntelMQ had the necessary permissions to access and write to the PostgreSQL database required additional configuration steps.

Troubleshooting Steps:

Connection Verification: Verified the database connection string by attempting to connect to the PostgreSQL database using a separate client.

Permission Review: Reviewed and granted the necessary privileges to the user running IntelMQ to access and write to the PostgreSQL database.

Log Examination: Examined the IntelMQ logs for any errors or warnings related to the database connection or writing data to the database.