Microsoft Sentinel is an excellent SIEM. It combines advanced analytics with significant customization capabilities. However, applying advanced analytics to your custom logs is hard.

This hackathon project gathered the best Sentinel content minds to resolve this issue. Devika, Ofer, Will, Edi, Jitesh, and Ofer joined forces to create powerful custom analytics in Sentinel. We wanted to create a solution using only Sentinel's content building blocks, such as KQL, Logic Apps, Analytic Rules, and Workbooks so that anyone (that's You!) can join and contribute.

Let's first look at the challenge at hand. KQL is powerful, and one can implement sophisticated analytics using only KQL. But there are two caveats:

First, analytic rules that build behavioral profiles each time they run are inefficient. In practice, analytic rules are limited to looking back only 14 days, and hence profiling only a fortnight, exactly because it is too resource-consuming to go back further.

Secondly, analytics is complicated, and using KQL for analytics is no exception. Trying to use KQL analytics capabilities is for the initiated.

So, how did we go about solving those challenges?

We have used Logic Apps to ensure efficiency by summarizing the data and generating predictions. Those are written back to the Sentinel workspace and are used by analytic rules to detect anomalies. And to make things simpler, all elements read their configuration from Watchlists, eliminating the need for editing Logic Apps or KQL queries to tweak your analytics.

Let's look at the details:

We start with Devika's Logic App, which runs on a schedule and summarizes new events from your selected data sources. The Logic App creates aggregative time bins, which are much more efficient for longer-term analysis. Using a Watchlist, you can decide which sources and fields to analyze. This is how you connect your custom logs to our solution.

The bins are then analyzed by a second set of Logic Apps that Will wrote, which create predictions. These scheduled Logic Apps must look back into history, but since they analyze aggregated data, they can do so in a performant manner.

To start with, we implemented two prediction algorithms. The first is based on time series decomposition, and the second on rare relationships.

The analytics rule Ofer created compares the predictions to the current data and generates anomalies. We use Sentinel alerts as anomalies by annotating them using a custom field. We also don't trigger an incident for each anomaly individually.

Those anomalies can be used in several ways. A workbook can be used to investigate anomalies. The anomaly can also be used alongside other indicators to trigger an incident. We have created a rule that aggregates our anomalies, which should serve only as an example. In a real world situation, those anomalies should be combined with other signals in Sentinel to generate incidents.

This is a work in progress. You can find what we have delivered so far on our GitHub repository at https://aka.ms/Sentinel-Custom-Analytics. Since it is all Sentinel content, you can use this as a starting point for your experimentation with Sentinel custom analytics.