KQL Query Cheat Sheet

This cheat sheet provides KQL queries to assist during the training labs.

Contact us
info@invictus-ir.com

> Querying Basics

Querying a Table

To query data, you first need to specify the table name you want to explore.

```
["Lab1-2-SignInLogs"]
```

Querying the table without any filters or operations will return all rows and columns from the **["Lab1-2-SignInLogs"]** table.

If you want to query data from multiple tables at once, you can use the union operator.

```
union ["Lab1-2-SignInLogs"], ["Lab1-3-AuditLogs"]
```

Using take to Limit the Number of Rows

The take operator is used to limit the number of rows returned in the query result. This is useful when you only want to see a subset of the data.

```
["Lab1-2-SignInLogs"]
| take 10
```

The take command retrieves and displays the first 10 rows from the ["Lab1-2-SignInLogs"] table.

Using project to Select Specific Columns

You can use the project operator to reduce the number of columns in your results. This is useful when you only want to display or analyze certain columns from a table.

```
["Lab1-2-SignInLogs"]
| project UserPrincipalName, IPAddress, TimeGenerated
```

The project command selects and displays only the columns UserPrincipalName, IPAddress, and TimeGenerated.

> Filtering Data

Where clause

Use the **where** operator to filter the data based on a condition. This allows you to focus on specific rows that match the criteria.

```
"Lab1-2-SignInLogs"]
| where Username == "adelev@bonacu.onmicrosoft.com"
```

This query will return all rows where the username is adelev@bonacu.onmicrosoft.com.

Search across data

The search operator is a powerful tool in KQL that allows you to search for a specific keyword or value across all columns of a table.

```
["Lab1-2-SignInLogs"]
| search "adelev@bonacu.onmicrosoft.com"
```

In this query, the search operator scans the ["Lab1-2-SignInLogs"] table for any occurrence of the string "adelev@bonacu.onmicrosoft.com". It will return all rows where the term is present in any column, not just specific columns.

Time Filtering

Use time filtering to limit the data to a specific time range. The where clause, combined with ago() or explicit time ranges, allows you to focus on data within a certain timeframe.

```
["Lab1-2-SignInLogs"]
| where TimeGenerated >= ago(7d)
```

This query filters the logs to only show data from the past 7 days. You can adjust the time range by changing 7d to other values .

Has and Contains operator

The has operator searches for partial matches in a specific column. It looks for the occurrence of a value or keyword within the data in a column.

```
["Lab1-2-SignInLogs"]
| where IPAddress has "192.168"
```

This query filters rows in the table where the IPAddress column contains the substring "192.168".

Similar to has, the contains operator searches for partial matches but is case-insensitive and more flexible..

> Counting and Summarizing

Counting rows with count

The count operator is used to return the total number of rows that match a specific condition. This is useful when you want to quickly get the number of occurrences for a certain event or criteria.

```
["Lab1-2-SignInLogs"]
| count
```

This query counts the total number of rows in the table.

Summarize Data

The summarize operator is used to group rows and perform aggregation functions (e.g., counting, averaging) on the data. It helps condense large sets of data into meaningful summaries.

```
["Lab1-2-SignInLogs"]
| summarize count() by Username
```

In this query, the summarize operator counts the number of sign-ins per Username in the Lab1-2-SignInLogs table, grouping the results by username.

> JSON Data

JSON Parsing

Use the parse_json() function to work with JSON-formatted data inside a table. It allows you to extract specific properties from JSON strings stored in columns.

```
["Lab2-1-UAL"]
| extend d = parse_json(AuditData)
| evaluate bag_unpack(d, 'AuditData_')
```

This query parses JSON data from the AuditData column and unpacks all key-value pairs into individual columns, each prefixed with AuditData_. It allows easier access to specific properties within the JSON data.

INVICTUS
INCIDENT RESPONSE