KQL Language

resources

https://learn.microsoft.com/en-us/microsoft-365/security/defender/advanced-hunting-query-language?view=o365-worldwide

https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/

https://learn.microsoft.com/en-us/training/modules/explore-fundamentals-kql/

https://aka.ms/lademo

https://www.youtube.com/@TenMinuteKQL

https://github.com/cyb3rmik3/KQL-threat-hunting-queries

---

https://learn.microsoft.com/en-us/azure/sentinel/kusto-overview

https://www.udemy.com/course/learn-kql-for-microsoft-sentinel/

https://cloudacademy.com/lab/introduction-to-kusto-query-language/

https://learn.microsoft.com/en-us/training/modules/write-first-query-kusto-query-language/

Data Types in KQL

- **bool (boolean):** Represents true or false values. Used for logical comparisons and conditions.
  - Example: `StormEvents | where EventType == "Tornado"`
- **datetime (date):** Represents instants in time. KQL datetime values typically include both date and time information.
  - Example: `StormEvents | where StartTime > datetime(2023-12-20)`
- **decimal:** Represents a 128-bit decimal number. Ideal for high-precision financial and scientific calculations.
  - Example: `T | summarize TotalProfit = sum(Price * Quantity)`
- **dynamic:** A flexible data type that can hold the following:
  - **Arrays:** Ordered lists of values of any data type.

    - **Dictionaries (property bags):** Key-value pairs where keys are strings and values can be any data type.
    - **Any of the other primitive scalar data types.**
    - Example: `print details = dynamic({"Name": "Alice", "Scores": [95, 80, 92]})`
- **guid (uuid, uniqueid):** Represents a 128-bit globally unique identifier.
    - Example: `T | where TraceId == guid("123e4567-e89b-12d3-a456-426614174000")`
- **int:** Represents a 32-bit signed integer.
    - Example: `AppUsage | where SessionCount > 10`
- **long:** Represents a 64-bit signed integer.
    - Example: `ServerLogs | summarize TotalBytes = sum(PayloadSize)`
- **real (double):** Represents a 64-bit floating-point number.
    - Example: `CpuMetrics | where LoadPercentage >= 85.0`
- **string:** Represents a sequence of Unicode characters (text).
    - Example: `UserInfo | where UserEmail contains "example.com"`
- **timespan (time):** Represents durations or time intervals.
    - Example: `Events | where TimeSinceLastUpdate > 1h`

Syntax

- // > comment
- ctrl + k + c > comment
- ctrl + k + u > uncomment
- where >> clause is used to filter records based on specified conditions
- take >> operator is used to limit the number of records (rows)
- cluster('help').database('SecurityLogs').Email >> KQL made from cluster contain DB contain table
- == >> exact match
- != >> does not include
- =~ >> matching for regex pattern
- contains >> operator for string matching
- has >> looks for full string but can use delimiters and is not case sensitive
- endswith >> to get the end of a string
- startswith >> to get the starting of the string
- to make it case sensitive add _cs >> startswith_cs or endswith_cs
- sort by >> sorts the result by time or hostname or what ever u want

- order by >> similar to sort by
- distinct >> used to retrieve unique values from a specified column in a table
- project >> display the rows we want only
- project-reorder >> To achieve reordering of columns
- instead of using two where user one where and (and)
- where * has "389" >> to search in all the records for 389
- limit >> same as take
- top >> same as take but it give u the top of the results // top 10 by TimeGenerated desc
- TimeGenerated >= ago(7d) >> to display a timestamp greater than 7 days
- TimeGenerated <= ago(4h) >> less than four hours // less than mean older
- between >> to display a the frame between two dates // where StartTime between (datetime(2007-07-27) .. datetime(2007-07-30))
- now >> display the time between date and NOW // where StartTime between (datetime(2007-07-27) .. now())
- now - date >> same as above but to display four hours earlier //// where StartTime between (datetime(2007-07-27) .. now() -4h))
- search >> used to perform full-text searches on string columns within a table // same as has
  - if u used * at the end of the sting it is same as startswith
  - if u used * before the string it is same as endswith
  - if u used *string* is it same as contain
  - u can search in a column //where ColumnName search "*pattern*"
  - search "185.125.190.23"
    | distinct $table   >> to display all the unique tables that contain this IP
- search in (table1, table2) "string"
- extend >> to create a new field that does not exist //
  Usage
  | extend GB=Quantity/1000
  | sort by GB desc
- getschema >> Produce a table that represents a tabular schema of the input.
- isnull(value) >> used to check whether a scalar value is the
- isempty(string) >> used to check if a value is an empty string or null
- print "KQL is \"cool\"!" >> KQL is "cool"!
- tostring() >> This function explicitly converts any data type to its string
- summarize >> It allows you to group rows based on specific criteria and then apply various aggregation functions // | summarize count() by Location, UserType

- makke_set() >> Creates a dynamic array of distinct values from a column within a group, used to create a dynamic array containing the **distinct values** from a specific column within a group
- min() >> used to find the **minimum value** within a group of data.
- max() >> used to find the **maximum value** within a group of data.
- tostring() >> used to **convert any data type** into a **string representation**

Hunting Queries

Microsoft 365 Defender

The following query will present email details that have been identified as suspicious after delivery.

let CompromizedEmailAddress = ""; // Insert the email address of the compromised email address
let Timeframe = 2d; // Choose the best timeframe for your investigation
let EmailInformation = EmailEvents
| where RecipientEmailAddress == CompromizedEmailAddress
| where Timestamp > ago(Timeframe)
| where DeliveryAction != "Blocked"
| project Timestamp, NetworkMessageId, SenderMailFromAddress, SenderFromAddress, SenderDisplayName, ThreatNames;
EmailInformation
| join (EmailPostDeliveryEvents
| where ThreatTypes != ""
| project Timestamp, NetworkMessageId, Action, ActionType, ActionTrigger, ActionResult, DeliveryLocation, ThreatTypes, DetectionMethods
) on NetworkMessageId
| sort by Timestamp desc

This query provides a daily breakdown, indicating the percentage of detections attributed to various security products

```
AlertInfo
| where TimeGenerated > ago(7d)
| summarize TotalAlertCount = count(),
            App_Governance = countif(ServiceSource == "App Governance"),
            AAD_Identity_Protection = countif(ServiceSource == "AAD Identity Protection"),
            Microsoft_365_Defender = countif(ServiceSource == "Microsoft 365 Defender"),
            Microsoft_Defender_for_Identity = countif(ServiceSource ==
```

```
"Microsoft Defender for Identity"),
            Microsoft_Defender_for_Cloud_Apps = countif(ServiceSource ==
"Microsoft Cloud App Security"),
            Microsoft_Defender_for_Office365 = countif(ServiceSource ==
"Microsoft Defender for Office 365"),
            Microsoft_Defender_for_Endpoint = countif(ServiceSource ==
"Microsoft Defender for Endpoint"),
            Microsoft_Data_Loss_Prevention  = countif(ServiceSource ==
"Microsoft Data Loss Prevention") by bin(TimeGenerated, 1d)
| extend App_Governance_percentage = todouble(round(App_Governance /
todouble(TotalAlertCount) * 100, 2))
| extend AAD_Identity_Protection_percentage =
todouble(round(AAD_Identity_Protection / todouble(TotalAlertCount) * 100,
2))
| extend Microsoft_365_Defender_percentage =
todouble(round(Microsoft_365_Defender / todouble(TotalAlertCount) * 100, 2))
| extend Microsoft_Defender_for_Identity_percentage =
todouble(round(Microsoft_Defender_for_Identity / todouble(TotalAlertCount) *
100, 2))
| extend Microsoft_Defender_for_Cloud_Apps_percentage =
todouble(round(Microsoft_Defender_for_Cloud_Apps / todouble(TotalAlertCount)
* 100, 2))
| extend Microsoft_Defender_for_Office365_percentage =
todouble(round(Microsoft_Defender_for_Office365 / todouble(TotalAlertCount)
* 100, 2))
| extend Microsoft_Defender_for_Endpoint_percentage =
todouble(round(Microsoft_Defender_for_Endpoint / todouble(TotalAlertCount) *
100, 2))
| extend Microsoft_Data_Loss_Prevention_percentage =
todouble(round(Microsoft_Data_Loss_Prevention / todouble(TotalAlertCount) *
100, 2))
| project TimeGenerated,
          App_Governance_percentage,
          AAD_Identity_Protection_percentage,
          Microsoft_365_Defender_percentage,
          Microsoft_Defender_for_Identity_percentage,
          Microsoft_Defender_for_Cloud_Apps_percentage,
          Microsoft_Defender_for_Office365_percentage,
          Microsoft_Defender_for_Endpoint_percentage,
          Microsoft_Data_Loss_Prevention_percentage
| render columnchart
```

This query displays alerts detected in all Defender security products and correlates each of them with MITRE ATT&CK techniques

```
AlertInfo
| where TimeGenerated > ago(14d)
| where isnotempty(AttackTechniques)
| extend Parsed = parse_json(AttackTechniques)
| mv-expand Parsed
| extend MITRE_ATTCK = tostring(Parsed)
| extend PackedData = strcat(format_datetime(TimeGenerated,'yyyy-M-dd
H:mm:ss'), " : ", AlertId, " : ", Title, " : ", ServiceSource)
| summarize MDE = make_set_if(PackedData, ServiceSource == "Microsoft
Defender for Endpoint"),
            MDO = make_set_if(PackedData, ServiceSource == "Microsoft
Defender for Office 365"),
            MDI = make_set_if(PackedData, ServiceSource == "Microsoft
Defender for Identity"),
            MDA = make_set_if(PackedData, ServiceSource in ("Microsoft Cloud
App Security", "App Governance")),
            Entra = make_set_if(PackedData, ServiceSource == "AAD Identity
Protection"),
            M365D = make_set_if(PackedData, ServiceSource == "Microsoft 365
Defender") by MITRE_ATTCK
| extend MDE_case = array_length(MDE)
| extend MDO_case = array_length(MDO)
| extend MDI_case = array_length(MDI)
| extend MDA_case = array_length(MDA)
| extend Entra_case = array_length(Entra)
| extend M365D_case = array_length(M365D)
| extend SUM = MDE_case + MDO_case + MDI_case + MDA_case + Entra_case +
M365D_case
| project MITRE_ATTCK, SUM, MDE, MDO, MDI, MDA, Entra, M365D
| order by SUM desc
```