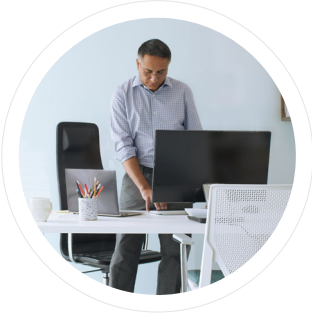
Microsoft Security

SC-200 Microsoft Security Operations Analyst

Ben McGee
11/20/2022

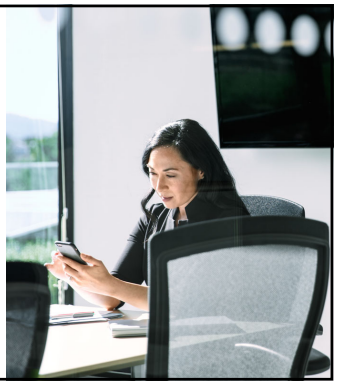


© Copyright Microsoft Corporation. All rights reserved.

1

Microsoft Security

Module 4: Create queries for Microsoft Sentinel using Kusto Query Language (KQL)



© Copyright Microsoft Corporation. All rights reserved.

2

Module agenda

- Construct KQL statements for Microsoft Sentinel
- Analyze query results using KQL
- Build multi-table statements using KQL
- Work with string data in using KQL statements

© Copyright Microsoft Corporation. All rights reserved.

3

© Microsoft Corporation

Lesson 1: Construct KQL statements for Microsoft Sentinel

4

Lesson introduction

After this lesson, you will be able to:

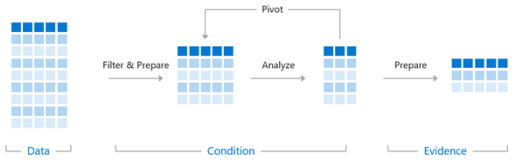
- Construct KQL statements
- Search log files for security events using KQL
- Filter searches based on event time, severity, domain, and other relevant data using KQL

5

The Kusto Query Language statement structure

A KQL query is a read-only request to process data and return results. The request is stated in plain text, using a data-flow model designed to make the syntax easy to read, write, and automate.

```
SecurityEvent | where EventID == "4626" | summarize count() by Account | limit 10
```



© Copyright Microsoft Corporation. All rights reserved.

6

© Microsoft Corporation

Use the let statement

```
let timeOffset = 7d;
let discardEventId = 4688;
SecurityEvent
| where TimeGenerated > ago(timeOffset*2) and TimeGenerated < ago(timeOffset)
| where EventID != discardEventId


let LowActivityAccounts =
    SecurityEvent
    | summarize cnt = count() by Account
    | where cnt < 10;

LowActivityAccounts | where Account contains "Mal"
```

© Copyright Microsoft Corporation. All rights reserved.

7

Use the search operator

```
search "err"

search in (SecurityEvent,SecurityAlert,A*) "err"
```

© Copyright Microsoft Corporation. All rights reserved.

8

Use the where operator

```
SecurityEvent
| where TimeGenerated > ago(1d)

SecurityEvent
| where TimeGenerated > ago(1h) and EventID == "4624"

SecurityEvent
| where TimeGenerated > ago(1h)
| where EventID == 4624
| where AccountType =~ "user"

SecurityEvent | where EventID in (4624, 4625)
```

© Copyright Microsoft Corporation. All rights reserved.

9

© Microsoft Corporation

Use the extend operator

```
let timeframe = 1d;

let DomainList = dynamic(["tor2web.org", "tor2web.com"]);

Syslog
| where TimeGenerated >= ago(timeframe)
| where ProcessName contains "squid"
| extend
  HTTP_Status_Code = extract("(TCP_(([A-Z]+)…-9]{3}))",8,SyslogMessage),
  Domain = extract("(([A-Z]+ [a-z]{4…Z]+ )([^ :\\/]*))",3,SyslogMessage)
| where HTTP_Status_Code == "200"
| where Domain contains "."
| where Domain has_any (DomainList)
```

© Copyright Microsoft Corporation. All rights reserved.

10

Use the order by operator

```
SecurityAlert
| where TimeGenerated > ago(7d)
| extend severityOrder = case (
    AlertSeverity == "High", 3,
    AlertSeverity == "Medium", 2,
    AlertSeverity == "Low", 1,
    AlertSeverity == "Informational", 0,
    -1)
| order by severityOrder desc
```

© Copyright Microsoft Corporation. All rights reserved.

11

Use the project operators

```
SecurityEvent
| project Computer, Account

SecurityAlert
| where TimeGenerated > ago(7d)
| extend severityOrder = case (
    AlertSeverity == "High", 3,
    AlertSeverity == "Medium", 2,
    AlertSeverity == "Low", 1,
    AlertSeverity == "Informational", 0,
    -1)
| order by severityOrder
| project-away severityOrder
```

Operator	Description
project	Select the columns to include, rename or drop, and insert new computed columns.
project-away	Select what columns from the input to exclude from the output.
project-keep	Select what columns from the input to keep in the output.
project-rename	Select the columns to rename in the resulting output.
project-reorder	Set the column order in the resulting output.

© Copyright Microsoft Corporation. All rights reserved.

12

© Microsoft Corporation

Lesson 2: Analyze query results using KQL

13

Lesson introduction

After this lesson, you will be able to:

- Summarize data using KQL statements
- Render visualizations using KQL statements

14

Use the summarize operator

```
SecurityEvent
| summarize count() by Process, Computer

let timeframe = 1d;
SigninLogs
| where TimeGenerated >= ago(timeframe)
| where ResultType == "50057"
| where ResultDescription =~ "User account is disabled. The account has been disabled by an administrator."
| summarize applicationCount = dcount(AppDisplayName) by UserPrincipalName, IPAddress
| where applicationCount >= 3
```

Function(s)	Description
count(), countif()	Returns a count of the records per summarization group
dcount(), dcountif()	Returns an estimate for the number of distinct values taken by a scalar expression in the summary group.
avg(), avgif()	Calculates the average of Expr across the group.
Max(), maxif()	Returns the maximum value across the group.
sum(), sumif()	Calculates the sum of Expr across the group.

© Copyright Microsoft Corporation. All rights reserved.

15

© Microsoft Corporation

Use the summarize operator to filter results

```
SecurityEvent
| where Computer == "SQL12.NA.contosohotels.com"
| summarize arg_max(TimeGenerated,*) by Computer

SecurityEvent
| where Computer == "SQL12.NA.contosohotels.com"
| summarize arg_min(TimeGenerated,*) by Computer
```

© Copyright Microsoft Corporation. All rights reserved.

16

Use the summarize operator to prepare data

```
SecurityEvent
| where EventID == "4624"
| summarize make_list(Account) by Computer


SecurityEvent
| where EventID == "4624"
| summarize make_set(Account) by Computer
```

© Copyright Microsoft Corporation. All rights reserved.

17

Use the render operator to create visualizations

```
SecurityEvent
| summarize count() by Account
| render barchart

SecurityEvent
| summarize count() by bin(TimeGenerated,
1d)
| render timechart
```

Visualizations
areachart
barchart
columnchart
piechart
scatterchart
timechart

© Copyright Microsoft Corporation. All rights reserved.

18

© Microsoft Corporation

Lesson 3: Build multi-table statements using KQL

19

Lesson introduction

After this lesson, you will be able to:

- Create queries using unions to merge results from multiple tables using KQL
- Merge two tables with the join operator using KQL

20

Use the union operator

```
SecurityEvent
| union SecurityAlert


union Security*
| summarize count() by Type
```

© Copyright Microsoft Corporation. All rights reserved.

21

© Microsoft Corporation

Use the join operator

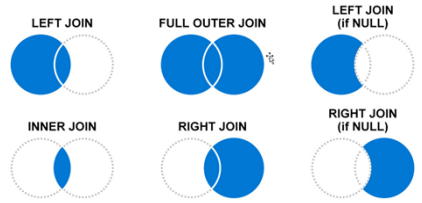
```
SecurityEvent
| where EventID == "4624"
| summarize LogOnCount=count() by EventID, Account
| project LogOnCount, Account
| join kind = inner (
    SecurityEvent
    | where EventID == "4634"
    | summarize LogOffCount=count() by EventID, Account
    | project LogOffCount, Account
) on Account
```

© Copyright Microsoft Corporation. All rights reserved.

22

Use the join operator (continued)

When joining tables, you use Join flavors to determine the joining behavior. It is essential to understand the impact of records on the left and right side based on the join flavor.



© Copyright Microsoft Corporation. All rights reserved.

23

Lesson 4: Work with string data using KQL statements

24

© Microsoft Corporation

Lesson introduction

After this lesson, you will be able to:

- Extract data from unstructured string fields using KQL
- Extract data from structured string data using KQL
- Create Functions using KQL

25

Extract data from unstructured string fields

Extract function:

```
let top5 = SecurityEvent
| where EventID == 4625 and AccountType == 'User'
| extend Account_Name = extract(@"^(.*\\)?([^@]*)(@.*)?$", 2, tolower(Account))
| summarize Attempts = count() by Account_Name
| where Account_Name != ""
| top 5 by Attempts
| summarize make_list(Account_Name);

SecurityEvent
| where EventID == 4625 and AccountType == 'User'
| extend Name = extract(@"^(.*\\)?([^@]*)(@.*)?$", 2, tolower(Account))
| extend Account_Name = iff(Name in (top5), Name, "Other")
| where Account_Name != ""
| summarize Attempts = count() by Account_Name
```

© Copyright Microsoft Corporation. All rights reserved.

26

Extract data from unstructured string fields (continued)

Parse function:

```
Event
| where RenderedDescription !has "LGIS" and RenderedDescription !has "LGIF"
| parse RenderedDescription with * "action_id:" Action:string
                                  " " *
| parse RenderedDescription with * "client_ip:" ClientIP:string
" permission" *
| parse RenderedDescription with * "session_server_principal_name:"
CurrentUser:string
" " *
| parse RenderedDescription with * "database_name:" DatabaseName:string
"schema_name:" Temp:string
"object_name:" ObjectName:string
"statement:" Statement:string
"." *
```

© Copyright Microsoft Corporation. All rights reserved.

27

© Microsoft Corporation

Extract data from structured string data

Parse dynamic fields:

```
AzureActivity
| project Properties_d.eventCategory
```

Work with JSON data:

```
SecurityAlert
| extend ExtendedProperties = todynamic(ExtendedProperties)
| extend ActionTaken = ExtendedProperties.ActionTaken
| extend AttackerIP = ExtendedProperties["Attacker IP"]
```

© Copyright Microsoft Corporation. All rights reserved.

28

Integrate external data

```
Users
| where UserID in ((externaldata (UserID:string) [
    @"https://storageaccount.blob.core.windows.net/storagecontainer/users.txt"
      h@"?...SAS..." // Secret token needed to access the blob
    ]))
| ...
```

© Copyright Microsoft Corporation. All rights reserved.

29

Create Parsers using functions

```
OfficeActivity
    | where TimeGenerated >= ago(30d)
    | where Operation == 'New-InboxRule’


// Save the query as a function named MailboxForward

MailboxForward
```

© Copyright Microsoft Corporation. All rights reserved.

30

© Microsoft Corporation

Knowledge check

Check your knowledge with the module quiz in your course viewer

31

Module 4, Lab 01 – Create queries for Microsoft Sentinel using Kusto Query Language (KQL)

32

Lab Exercises for module 4

1. Create queries for Microsoft Sentinel using Kusto Query Language (KQL)

© Copyright Microsoft Corporation. All rights reserved.

33

© Microsoft Corporation

Microsoft Security

© Copyright Microsoft Corporation. All rights reserved.

34

© Microsoft Corporation