Welcome to the Azure Sentinel webinar

Microsoft

We will start at **2-3 minutes after** the scheduled time to accommodate those still connecting.

Questions? Feel free to type them in the instant message window at any time. Note that any questions you post will be public. You have the option to post questions anonymously. After the webinar, you can ask questions at https://aka.ms/AzureSentinelCommunity.

This webinar is being **recorded**. We'll post the recordings to our community forums at https://aka.ms/SecurityWebinars.

Please give us your **feedback** on this webinar at https://aka.ms/SecurityCommunityWebinarFeedback.

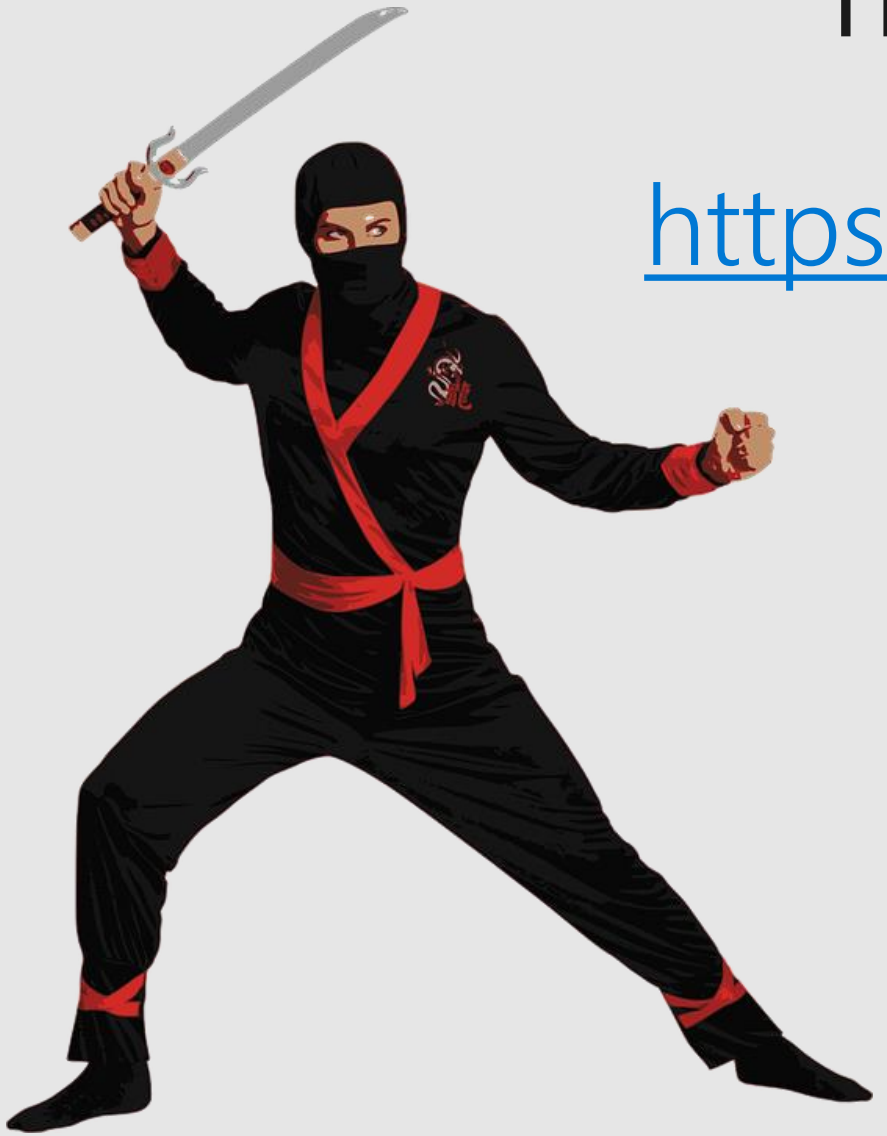
Join our Community: https://aka.ms/SecurityCommunity

Microsoft

Azure Sentinel

KQL for Azure Sentinel Hands-on Lab

Ofer Shezaf



This webinar is a part of a series!

https://aka.ms/SentinelNinjaTraining

for the full season

Your KQL journey

KQL Pluralsight course,
Pluralsight Advanced KQL course

This module: KQL for Sentinel hands-on lab

Ninja Training Module 8: Writing rules

Ninja Training Module 11: Use cases

Exercise yourself

https://aka.ms/lademo

(requires any MS account)

Home > Azure Sentinel | Logs

Azure Sentinel | Logs

Selected workspace: 'CyberSecurityDemo'

New Query 1*

Example queries | Query explorer

CyberSecurityDemo | Run | Time range : Last 24 hours | Save | Copy link | New alert rule | Export | Pin to dashboard | Prettify query

Implicit time filter

Tables | Filter

Search

Group by: Solution Filters: not selected

- Favorites
- BehaviorAnalyticsInsights
- Change Tracking
- ContainerInsights
- DNS Analytics (Preview)
- LogManagement
- Network Performance Monitor
- Office 365 Analytics (Preview)
- Security and Audit
  - CommonSecurityLog
    - t Activity (string)
    - t AdditionalExtensions (string)
    - t ApplicationProtocol (string)
    - t CommunicationDirection (string)
    - t Computer (string)
    - t DestinationDnsDomain (string)
    - t DestinationHostName (string)
    - t DestinationIP (string)
    - t DestinationMACAddress (string)
    - t DestinationNTDomain (string)

Tables

Typed columns

```
SecurityEvent
| where EventID == 4625
```

Queries

Results | Chart | Columns | Add bookmark | Display time (UTC+00:00)

Completed. Showing partial results from the last 24 hours. 00:00:06.015 10,000 records

umn header and drop it here to group by that colum

Column chooser

TimeGenerated [UTC]	Account	AccountType	Computer	EventSourceName	Chanr
4/22/2020, 10:31:04.483 AM	\OSLADMIN	User	VictimPC.Contoso.Az...	Microsoft-Windows-Security-Auditing	Securi
4/22/2020, 10:31:04.760 AM	\OUJADM	User	VictimPC.Contoso.Az...	Microsoft-Windows-Security-Auditing	Securi
4/22/2020, 10:31:04.903 AM	\SVCADM	User	VictimPC.Contoso.Az...	Microsoft-Windows-Security-Auditing	Securi

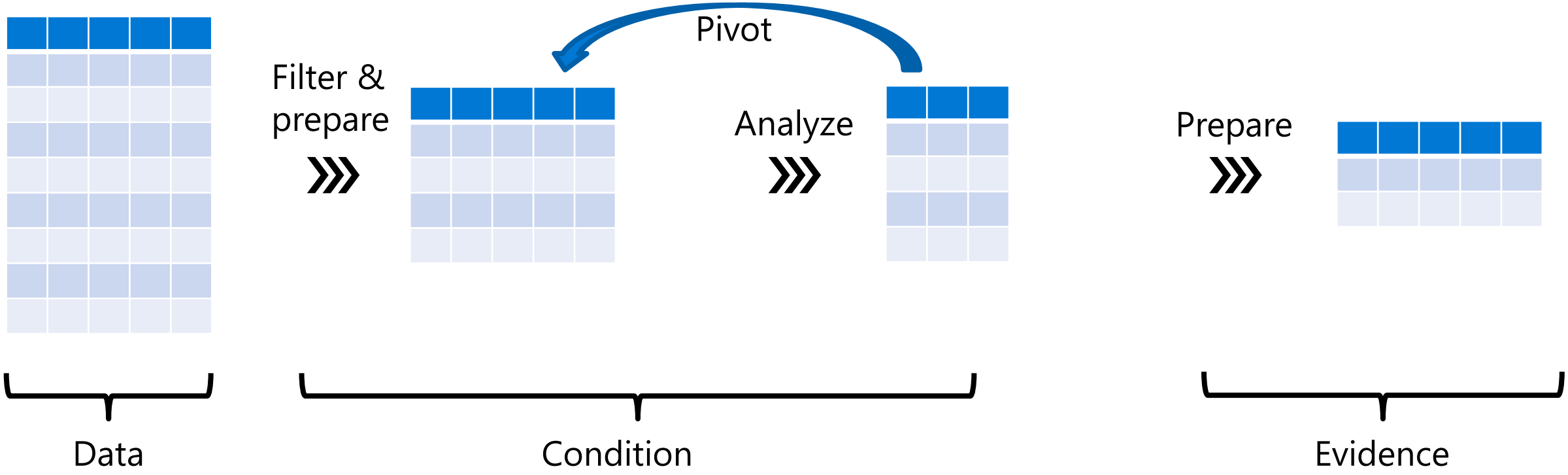
Results

TenantId	ab86c959-1ba3-495c-a00d-ced30d8825d3
TimeGenerated [UTC]	2020-04-22T10:31:04.903Z
SourceSystem	OpsManager
Account	\SVCADM
AccountType	User
Computer	VictimPC.Contoso.Azure
ourceName	Microsoft-Windows-Security-Auditing
el	Security
Task	12544

Page 1 of 200 | 50 items per page | 1 - 50 of 10000 items

Understanding the pipe

```
SecurityEvent | where EventID == "4624" | summarize count() by Account | top 10 by _count
```



Filter & Prepare

Flow


Choose your input tables
Narrow down on the events of interest
Parse & prepare fields

Choose a table

Just use the table name:

- Standard tables
- Custom tables

Or

- "union" - query multiple tables
- "externaldata" – query a table in an external file
- "datatable" – query a static table, for example for testing
- Stored functions – use a pre-prepared and parsed virtual table

See advanced topics

‘where’ operator

Filters a table to the subset of rows that satisfy a predicate.

Syntax: *T | where Predicate*

Examples: *SecurityEvent | where TimeGenerated > ago(1d)*

*SecurityEvent | where * contains “Kusto”*

Operators:

- **String** : ==, has, contains, startswith, endswith, matches regex
- **Numeric/Date**: ==, !=, <, >, <=, >=
- **Lookup**: in, !in, has_any
- And many more!

Supports **and**, **or**, and **not()**

‘where’ exercise

Start with filtering by time

Time range type

Relative times

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

SecurityEvent
| where TimeGenerated > ago(1h) and EventID == 4624 // Successful logon

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

Case insensitive

Lists can be dynamic

SecurityEvent | where EventID in (4624, 4625)

AzureNetworkAnalytics_CL | where ipv4_is_match(DestIP_s, "10.0.0.0/8")

Breadth of operators

‘search’ operator

Easy to use. Inefficient. Use interactively, but not in content.

Syntax: *[T |] search “string” [in (Tables)]*

Examples: *search “10.1.5.5”*

SecurityEvent | where TimeGenerated >= ago(1h) | search “Guest”

- “T |” and “in (Tables)” are optional. With no table specified will search all tables.
- The “$table” field will include the table name if a multi-table search.

‘search’ shortcuts

Syntax	Meaning (equivalent where)
search "err"	union * | where * has "err"
search in (T1,T2,A*) and "err"	union T1,T2,A* | where * has "err"
search col:"err"	union * | where col has "err"
search col=="err"	union * | where col == "err"
search "err*"	union * | where * hasprefix "err"
search "Lab*PC"	union * | where * matches regex @"\bLab\w*PC\b"
search "abc" and ("def" or "hij")	union * | where * has "abc" and (* has "def" or * has hij")

‘extend’ operator

Create calculated columns and append them to the result set

Syntax: *T | extend ColumnName [= Expression] [, ...]*

Example: *SecurityEvent | extend ComputerNameLength = strlen(Computer)*

- The new added column is not stored.
- To only change a column name, use ‘project-rename’.
- Expression capabilities are endless.
- Used for parsing.

‘extend’ exercise

```
Perf
| where CounterName == "Free Megabytes"
| where InstanceName == "C:"
| extend FreeKB = CounterValue * 1000
| extend FreeGB = CounterValue / 1000

SecurityEvent | where EventID in (4624, 4625)
| extend rgroup = extract("resourcegroups/(.*)/providers",1,_ResourceId)

… | extend rgroup = split(_ResourceId, "/",4)[0]
… | parse _ResourceId with "/subscriptions/" sub “/resourcegroups/" rgroup
"/providers" *
```

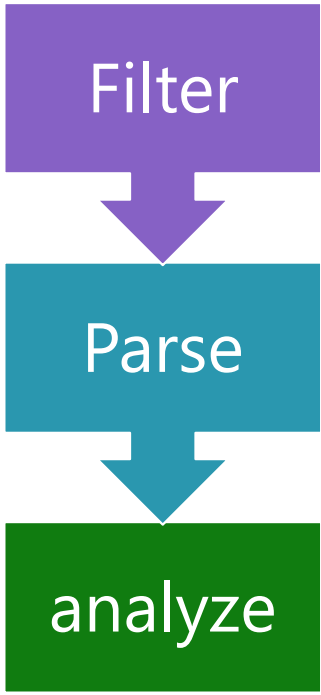
Use “extract” to parse a value

or “split”, "extract_all” or “parse”

A real-world example: tor usage detection

Use “let” to better organize queries

```
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)
```



https://github.com/Azure/Azure-Sentinel/blob/master/Detections/Syslog/squid_tor_proxies.yaml

Lab #1: filtering

Find all Windows logon events starting 2 weeks ago until 1 week ago that occurred on a computer with name which starts with "App".

Hints and guideline:

- Windows security events are stored int the table "SecurityEvent"
- The logon event id is 4624. What is the name of the field which contains the event ID?
- What is the name of the field which represents the computer name?
- What should be the order of the commands, but better performance?

Lab #1 solution, in steps

```
// Find all Windows logon events starting 2 weeks ago until 1 week ago that occurred on a computer
with name which starts with “App”

SecurityEvent | limit 100 // Find relevant fields: Activity, EventID, Computer

SecurityEvent | summarize by Activity // find the Event signaling login

SecurityEvent
| where TimeGenerated between (ago(14d)..ago(7d)) // start with the time filter
| where EventID == "4624"
| where Computer startswith "App" // case insensitive
        // This is the solution, but there are so many results

SecurityEvent
| where TimeGenerated between (ago(14d)..ago(7d))
| where EventID == "4624"
| where Computer startswith "App"
| summarize count() by Computer
        // so let’s count per computer
```

Analyze

‘summarize’ command

Produces a table that aggregates the content of the input table.

Syntax: *T | summarize Aggregation [by Group Expression]*

Examples: *SecurityEvent | summarize count() by Computer*

- Simple aggregation functions: count(), sum(), avg(), min(), max(),
- Advanced functions: arg_min(), arg_max(), make_list(), countif()

‘summarize’ exercise

```
WindowsFirewall
| where CommunicationDirection == "SEND"
| where FirewallAction == "ALLOW"
| summarize dcount(SourceIP)
```

Count distinct IP addresses for selected data set. Returns a single value.

```
SecurityEvent
| where TimeGenerated > ago(1h)
| where EventID == 4624
| summarize count() by AccountType, Computer
```

Count logins by user and computer

AccountType	Computer	count_
Machine	DC11.NA.contosohotels.com	320
Machine	DC10.NA.contosohotels.com	390
Machine	SQL00.NA.contosohotels.com	30
Machine	DC21.NA.contosohotels.com	374
Machine	DC00.NA.contosohotels.com	504

Note the default column name. Use c=count() to override

Lab #2: analysis

Find how many times each process ran per computer

Hints and guideline:

- Event 4688 logs process creation.
- Which field represent the processes created and which the computer on which it was ran?

Lab #2 solution, in steps

```
// Find how many times each process ran per computer

SecurityEvent | summarize by Activity // Let’s find the event that includes process names

SecurityEvent | where EventID == "4688" | limit 10
        // find the relevant field, in this case "Process"

SecurityEvent
| where EventID == "4688"
| summarize count() by Process, Computer
```

Variants and add-ons to summarize

Summarize shortcuts

```
SecurityEvent | distinct Computer, Account
SecurityEvent | where EventID == 4624 | count
```

Also useful

```
SecurityEvent | where EventID == 4624 | order by Account
SecurityEvent | top 10 by TimeGenerated desc
```

‘order by’ exercise

```
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
```

Note use of ‘case’. Last input value (-1) is the default.

‘project-away’ removes unneeded fields from the result set

‘summarize’ as a filter: arg_min(), arg_max()

Filter out top or bottom rows. Essentially “top by”.

```
WindowsFirewall
| where TimeGenerated > ago(7d)
| summarize arg_max(TimeGenerated, *) by SourceIP
```

Implies returning the entire event, even if part of ‘summarize’

Quiz #1

What is the difference between the following queries?

```
SecurityEvent
| summarize arg_max(TimeGenerated, *) by Account
| where EventID == "4624"
| count

SecurityEvent
| where EventID == "4624"
| summarize arg_max(TimeGenerated, *) by Account
| count
```

Quiz #1: solution

```
// count the accounts for which the last activity was a login
SecurityEvent
| summarize arg_max(TimeGenerated, *) by Account
| where EventID == "4624"

// count the number of Accounts which logged in
SecurityEvent
| where EventID == "4624"
| summarize arg_max(TimeGenerated, *) by Account
```

Lab #3: analysis

Find how many source IPs from which traffic has been dropped by Windows more than 10 times in the last 7 days.

Hints and guideline:

- Connections to Windows machines are collected in the “WindowsFirewall”
- What is the name of the field which specifies traffic direction and can help determine inbound connections?
- What is the name of the field which specifies the firewall action and can help identify dropped connections?

Lab #3 solution, in steps

```
// Find how many source IPs from which traffic has been dropped by Windows more than 10 times in
the last 7 days.

WindowsFirewall | limit 10
        // Identify fields: CommunicationDirection, FirewallAction, SourceIP

WindowsFirewall | summarize by CommunicationDirection, FirewallAction
        // What are the possible values?

WindowsFirewall
| where TimeGenerated > ago(7d)
| where FirewallAction == "DROP" and CommunicationDirection == "RECEIVE"
| summarize count() by SourceIP
| where count_ > 10
        // using implicit naming, using 'c=count()' can explicitly name
```

A real-world example: password spray detection

```
let timeframe = 1d;
let threshold = 3;
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 >= threshold
```

Filter failed login attempts to disabled accounts

Summarize distinct applications attempted per username and source IP

Determine if over a threshold

https://github.com/Azure/Azure-Sentinel/blob/master/Detections/SigninLogs/DisabledAccountSigninsAcrossManyApplications.yaml

Prepare

‘project’ operator

Select the columns to include, rename or drop, and insert new computed columns.

Syntax: *T | project ColumnName [= Expression] [, ...]*

Example: *SecurityEvent | project TimeGenerated, Computer*

‘| project-away’ – Removed specified column/s.

‘| project-rename’ – Rename specified column/s.

‘project’ exercise

```
SecurityEvent
| project IsImportant = iff(Computer contains "CEO", true, false)
```

An “if” function similar to Excel’s.

‘summarize’ to prepare: make_list(), make_set():

Keep all values of a summary group as a list:

- Ordered (make_list)
- Unique values (make_set)

Use to:

- Display all values to the user
- Create a lookup list

```
SecurityEvent
| summarize make_set(Account) by Computer
```

Back to password spray detection

```
let timeframe = 1d;
let threshold = 3;
SigninLogs
| where TimeGenerated >= ago(timeframe)
| where ResultType == "50057"
| where ResultDescription =~ "User account is disabled. The account has be
  disabled by an administrator."
| summarize StartTime = min(TimeGenerated), EndTime = max(TimeGenerated),
  count(), applicationSet = make_set(AppDisplayName),
  applicationCount = dcount(AppDisplayName), by UserPrincipalName, IPAddress
| where applicationCount >= threshold
| extend timestamp = StartTime, AccountCustomEntity = UserPrincipalName,
IPCustomEntity = IPAddress
```

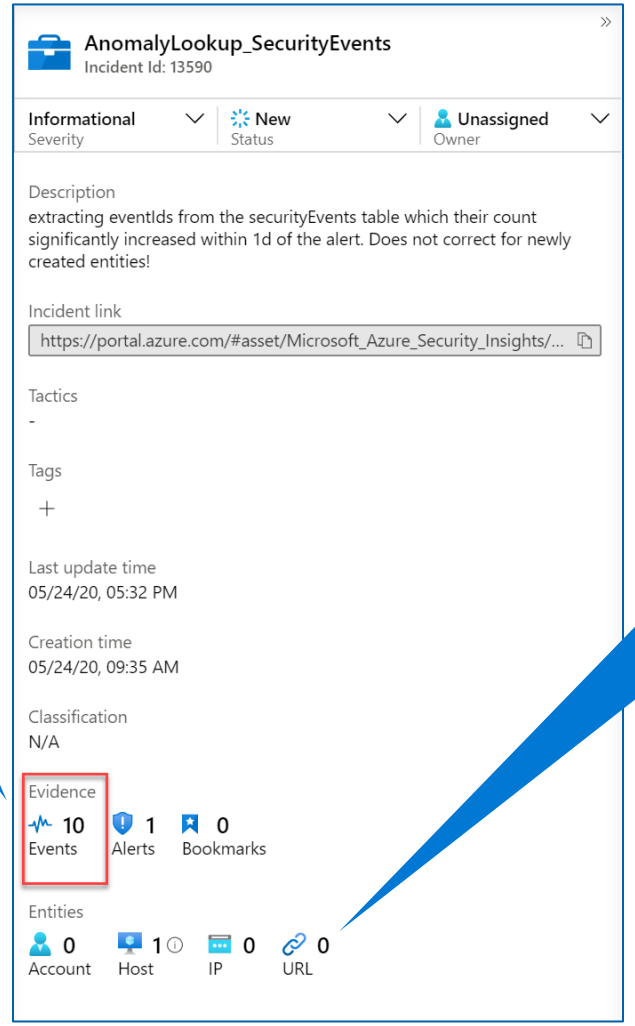
Keep essential data from the raw events for the analyst

Assign standard properties for later use, including entities

https://github.com/Azure/Azure-Sentinel/blob/master/Detections/SigninLogs/DisabledAccountSigninsAcrossManyApplications.yaml

Query output and Azure Sentinel incidents

Query output is available as events



Query output designated using standard entity fields is available as entities

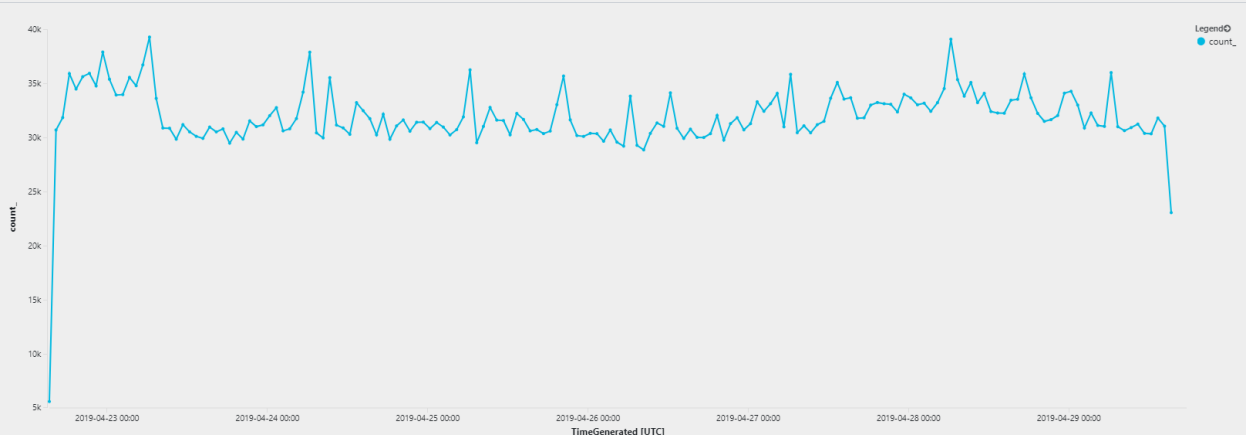
Visualize

‘summarize’: bin and time series

Bin is essentially the ‘floor’ function. It is very useful in summarize operations to creating time series.

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

Can create multiple overlaying charts by aggregating additional field



Render operator

Generates a visualization of the query results.

Syntax: *T* | *render Visualization [with (PropertyName = PropertyValue [, ...])]*

Supported visualizations:

- Areachart
- Barchart
- Columnchart
- Piechart
- Scatterchart
- timechart

‘bin’ exercise

```
SecurityEvent
| where TimeGenerated > ago(7d)
| summarize count() by bin(TimeGenerated, 1d)

VMConnection
| summarize count() by SourceIp | sort by count_ desc | render barchart
```

Lab #4: visualization

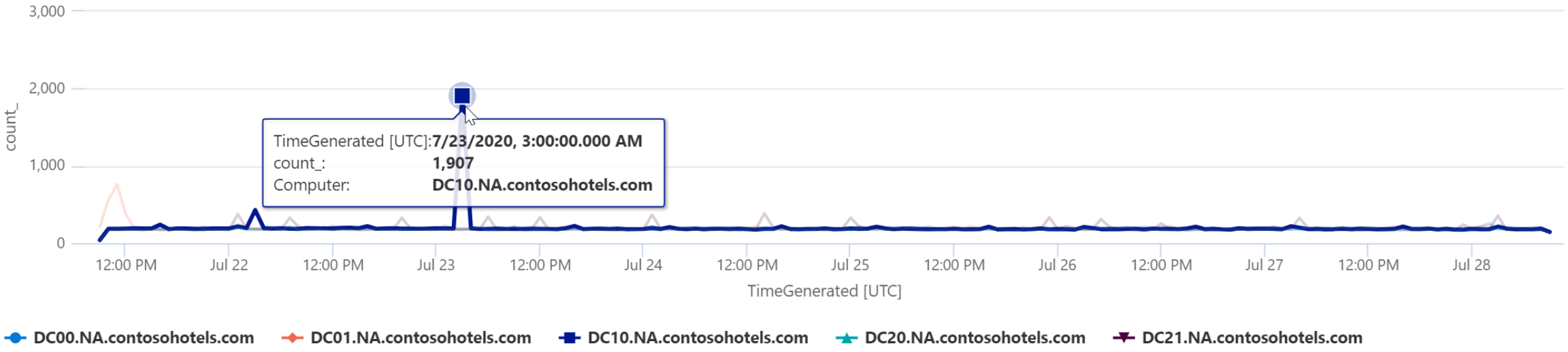
Chart the rate of process creation on all domain controllers.

Hints and guideline:

- Process creation is Windows event 4688
- Domain controller names start with “DC”
- Create multiple charts by aggregating additional more than one field

Lab #4 solution

```
// Chart the rate of process creation on all domain controllers.

SecurityEvent
| where Computer startswith "DC"
| where EventID == "4688" | summarize count() by Computer, bin(TimeGenerated, 1h)
| render timechart
```



Lab #5: visualization

1. *Render graph of allowed vs dropped connections over the last 7 days, use alias for the legend ("Allowed", "Dropped")*
2. *Render the ratio*

Hint: Check if the aggregation function countif can help.

Lab #5 solution

```
// Render graph of allowed vs dropped connections over the last 7 days, use alias for the legend
(“Allowed”, “Dropped”)
WindowsFirewall
| where TimeGenerated > ago(7d)
| where CommunicationDirection == "RECEIVE"
| summarize
        Dropped=countif(FirewallAction == "DROP"),
        Allowed=countif(FirewallAction == "ALLOW")
        by bin(TimeGenerated, 1h)
| render timechart

// Render the ratio
- Dropped=countif(FirewallAction == "DROP"),
- Allowed=countif(FirewallAction == "ALLOW")
+ Ratio=countif(FirewallAction == "DROP")/countif(FirewallAction == "ALLOW")
```

Advanced topics

‘let’ statement: declare and reuse variables

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

Note the semicolon at the end of the ‘let’ statement

‘let’ statement: declare dynamic tables or lists

```
let suspiciousAccounts = datatable(account: string) [
        @"\administrator",
        @"NT AUTHORITY\SYSTEM“
];
SecurityEvent | where Account in (suspiciousAccounts)

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

Declare a static table using the datatable operator

A one field table can be used as a list

Declare a table “view” which is a result of a query

‘materialize’ statement

Use with ‘let’ to cache and resuse the results of a query rather than run the query multiple times. Faster and ensures the same results are used.

Example-

```
let LowActivityAccounts =
 materialize(SecurityEvent
        | summarize cnt = count() by Account
        | where cnt < 10);
LowActivityAccounts
| where Account contains "Mal"
| union (LowActivityAccounts | where Account contains "Rep")
```

A simple ‘or’ would have worked instead of the ‘union’, but for the example sake…

‘union’ operator

Takes two or more tables and returns the rows of all of them.

Example:

```
SecurityEvent
| union (WindowsFirewall | where CommunicationDirection == "RECEIVE")
```

- kind=inner(common columns), outer (all columns- default)
- Supports wildcard to union multiple tables (union Security*)

Lab #6: union

Find the ratio of alerts (in the SecurityAlert table) to events (in the SecurityEvent table) broken by day for the last week

Lab #6 solution

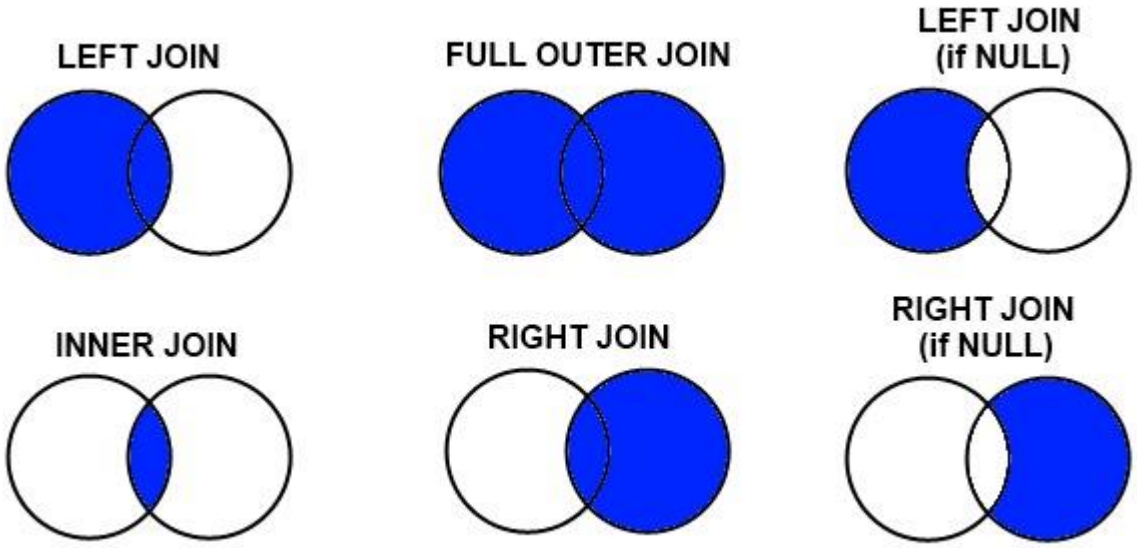
```
// Find the ratio of alerts (in the SecurityAlert table) to events (in the SecurityEvent table)
broken by day for the last week

SecurityAlert | union SecurityEvent
| where TimeGenerated > ago(7d)
| summarize
        // Type is s built in field which is set in each record to the table it originated from
        SecurityAlerts = countif(Type == "SecurityAlert"),
        SecurityEvents = countif (Type == "SecurityEvent")
        by bin (TimeGenerated, 24h)
| extend Ratio = SecurityAlerts * 1.0 / SecurityEvents
| project SecurityEvents , SecurityAlerts , Ratio
```

‘join’ operator

Merge the rows of two tables to form a new table by matching values of the specified column(s) from each table.

Syntax: LeftTable | join [JoinParameters] (RightTable) on Attributes

Example: *SecurityEvent* | *join (SecurityAlert* | *where Severity > 3) on Account*



Working with JSONs and arrays

- todynamic()- Convert the string to ‘dynamic’, a value of JSON type. Use either of these to refer to a field:
  - JsonField.Key
  - JsonField["Key"]

- mv-expand – Duplicate records, creating copies each one with one value of a JSON array. Easiest way to process JSON arrays.
- mv-apply – Apply a query to each value in an array.

JSON exercise

ExtendedProperties looks like a JSON but is a string and requires the use of todynamic

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

SecurityAlert
| mv-expand entity = todynamic(Entities)

SecurityAlert
| mv-apply entity = todynamic(Entities) on (
      where entity.Type == "account"
      | extend account = strcat (entity.NTDomain, "\\", entity.Name))
```

Lab #7

Show for each account that has alerts, how many alerts and which Security Events types it had in the last 7 days

Hints and guideline:

- Alerts in the SecurityAlert table keeps entities in a JSON array string. What is the name of this field?
- Account is just one of the possible entity types in the array.
- Make sure that account has the same format in both the alerts and events table.

Lab #7 solution

```
// Show for each account that has alerts, how many alerts and which Security Events types it had in
the last 7 days

// run parts of the query, adding a line at the time, to learn more
SecurityAlert
| mv-expand entity=todynamic(Entities) // mv-expand duplicate events: for each value in “Entities”,
you get a duplicate with “entity” set to the value
| where entity.Type == "account"
| extend Account = strcat(entity.NTDomain, "\\", entity.Name)
| summarize dcount(SystemAlertId) by Account
| join kind=leftouter (SecurityEvent | summarize make_set(EventID) by Account) on Account
```

Additional Links

Use the open to use KQL playground to exercise the labs

Consult with the:

- KQL documentation
- Pluralsight KQL course
- Pluralsight Advanced KQL course
- KQL cheat sheet

Q&A

Question	Answer
How do I create queries for source X or attach Y (KeyVault, Office 365 DLP, Conditional Access, Azure Information Protection, Unauthorized logins, Storage accounts, Data exfiltration, DNS attacks, SMB attacks, NTLMv2 and so on)	This Webinar did not focus on specific query building but rather on writing queries in general. To learn more about specific queries, look into module 11 of the Azure Sentinel Ninja training (Implementing Use Cases) as well as the Azure Sentinel GitHub.
Is there any way (shortcut) to comment several lines in KQL?	Unfortunately, not.
Is there documentation available for tables and field? like the purpose of table, field details etc.	See Azure Monitor schema reference
Is slack integration using webhooks with Azure Sentinel feasible?	To integrate Slack into your playbook, use the Logic App Slack connector here and the sample playbook here.
Do you know of KQL command to output the “dayofweek” as the full text-based day name, e.g. "Tuesday"? Only way I have been able to solve this is via case command.	You could use let to define an array (a list) of strings days, and extend: let days=dynamic(["Sunday", "Modayn", "Tuesday"]); print a=(dayofweek(datetime(2020-06-01)) == 1d) | extend day=days[a]
How can I delete/purge specific data?	See Manage PII management delete data from your workspaces
How can we iterate over all the available tables (for example to see lines that match in the last day)?	Use union * like this: union * | summarize count() by Type
Does KQL support external API request for enriching the data?	You will have to use Logic App to access the external API on a scheduled basis and populate a custom table.
Another question, do you know what the length limit is of a datatable?	A query, including the datatable, is limited to 2MB

Q&A

Question	Answer
Which database types does KQL has support for external queries? (only MS SQL?)	KQL cannot query external databases, only external files.
From performance perspective, which version is more efficient: *| where a == "b" and c > 4* or *| where a == "b" | where c > 4*	Even on huge dataset these are equivalent. For best practices see this page.
Does list work as "OR"?	The "in" operator is similar to a long list of ORs, though much more efficient.
Is there a keyboard shortcut for the "Run" button?	Yes, Shift-Enter
Is there a cheat sheet to commonly used searches and syntax?	See here.
Does KQL supports external file as an input to process a query?	Yes, see implementing lookups with Azure Sentinel.
Where we can see the RAW event?	This would depend on the specific source
Can we combine "case" and "matches regex" together?	Yes. the "matches regex" is a valid case predicate.
What's the difference between | project-rename and assigning a name to the column by using | project MyName=SomeColumn?	"project" creates a copy of the column (output would have both original and new names). project-rename will only has the new name. The latter is much more efficient.
How we can get the fields in each log stored in Sentinel? For example how we know there is field called computer or activity etc. to filter?	Writing rules indeed requires understanding the schema used by Azure Sentinel. You can find documentation for for key Microsoft and 3rd party sources and for most other Azure sources.
Is it better to pipe where or to use "and"?	There is no significant performance difference. Go with readability.

Q&A

Question	Answer
what is the best way to filter an exact time frame, i.e. using specific timestamps rather than the ago() function?	you can filter for exat time with datetime() or todatetime(). See date/time operations.
Is there any certain standard to write KQL query for better performance?	See Optimize log queries in Azure Monitor
Is there any way to get the event ID for the description that we are looking for ?	For Windows Security events I will consult the Microsoft Advanced security auditing FAQ
From where can I learn the advance level KQL queries such as: multiple table to be search in one query or how to join multiple columns?	Module 7 of the Azure Sentinel Ninja training, which includes this Webinar, contains several useful links.
How to convert Log analytics query to Analytical rule?	When you edit a query in LA\Sentinel click the [+ new alert rule]. You might also want to go through Module 8 of the Azure Sentinel Ninja training which covers rule writing using KQL.
Can the chart can be pinned to the centalized dashbaord?	You can pin charts to an Azure dashboard. Note that those are not used from within Azure Sentinel but can be accessed in the Azure portal. As an alternative, you can copy the query and use it in a workbook, which is accessible from Sentinel.
Can you elaborate on the mv-expand command? is the mv-expand somehow similar to bag_unpack function? or what's the differences?	bag_unpack takes key/value pairs in a dynamic object and create fields in the current event. mv-expand takes a list of values and create *multiple* events from them. This blog post might be useful.
How to extract and create a column for the specific fields from the extended properties?	***SecurityAlert | extend Countries = tostring(parse_json(ExtendedProperties).Countries)*** You can build this automatically in the UI. See details here: XX
how i can filter based on mimikatz.exe	***SecurityEvent | where EventID == 4688 | where CommandLine contains "mimikatz"***
how i can filter based on mimikatz.exe filename only ?	***SecurityEvent | where EventID == 4688 | where Process == "mimikatz.exe"***

Thank You for Joining Us!

Microsoft

Recordings will be posted to our community forums at https://aka.ms/SecurityWebinars.
Teams Live Event currently doesn't support audio capability for the audience. Closed captions are available during the webinar and with the recordings.

You can ask additional **questions** at https://aka.ms/AzureSentinelCommunity.

Please give us your **feedback** on this webinar at https://aka.ms/SecurityCommunityWebinarFeedback.

Join our Community: https://aka.ms/SecurityCommunity

SIEM Shift: How the Cloud is Transforming Security Operations
https://azure.microsoft.com/en-us/resources/why-companies-are-migrating-siem-to-the-cloud/

For any questions or comments on our documentation (https://docs.microsoft.com)
contact directly at MSsecuritydocs@microsoft.com