Azure Sentinel management using PowerShell

Kaido Järvemets

Microsoft MVP: Enterprise Mobility, MCT, Security+

Updated: 07.01.2020

Contents

Introduction

Now that we have an official PowerShell module for Azure Sentinel, we can use PowerShell with Azure Sentinel. In this small ebook, I will show you how to manage Azure Sentinel with the native cmdlets. Just to point out, that's the first version, and in version 0.1.0, we have these cmdlets:

Incident management:

- Get-AzSentinelIncident
- New-AzSentinelIncident
- Get-AzSentinelIncidentComment
- New-AzSentinelIncidentComment
- New-AzSentinelIncidentOwner
- Remove-AzSentinelIncident
- Update-AzSentinelIncident

Alert Rule Management:

- Get-AzSentinelAlertRule
- Get-AzSentinelAlertRuleAction
- Get-AzSentinelAlertRuleTemplate
- New-AzSentinelAlertRule
- New-AzSentinelAlertRuleAction
- Remove-AzSentinelAlertRule
- Remove-AzSentinelAlertRuleAction
- Update-AzSentinelAlertRule
- Update-AzSentinelAlertRuleAction

Bookmark Management

- Get-AzSentinelBookmark
- New-AzSentinelBookmark
- Remove-AzSentinelBookmark
- Update-AzSentinelBookmark

Connector Management

- Get-AzSentinelDataConnector
- New-AzSentinelDataConnector
- Remove-AzSentinelDataConnector
- Update-AzSentinelDataConnector

Requirements

- PowerShell modules
  - AzureAD
  - Az.SecurityInsights
  - Az
- Azure Sentinel activated on your Log Analytics workspace
- Permissions

Every Azure Sentinel cmdlet requires us to specify **ResourceGroupName** and **WorkspaceName** parameters. To simplify that part, we can define a hash table with the needed information. Later we can reference that hash table. You can read more about that method from here - [about_Splatting - PowerShell | Microsoft Docs]

So let's get started.

The first step is to install these three PowerShell modules from the PowerShell Gallery

- **Install-Module -Name Az.SecurityInsights -Verbose -Force**
- **Install-Module -Name AzureAD -Verbose -Force**
- **Install-Module -Name Az -Verbose -Force**

Please remember that you need administrative permissions to install these.

The next step is to make a connection to your Azure environment using the **Connect-AzAccount** cmdlet. You can read more about **Connect-AzAccount** from here - [Connect-AzAccount (Az.Accounts) | Microsoft Docs]

If you have access to different subscriptions, then you may need to change the subscription. To achieve that, just run the **Get-AzSubscription** cmdlet, copy the subscription **ID** where you have the Azure Sentinel workspace and then run **Set-AzContext** cmdlet like this:

- Set-AzContext -Subscription %**MySubcriptionID%**
  - **%MySubcriptionID%** should be replaced with the actual ID

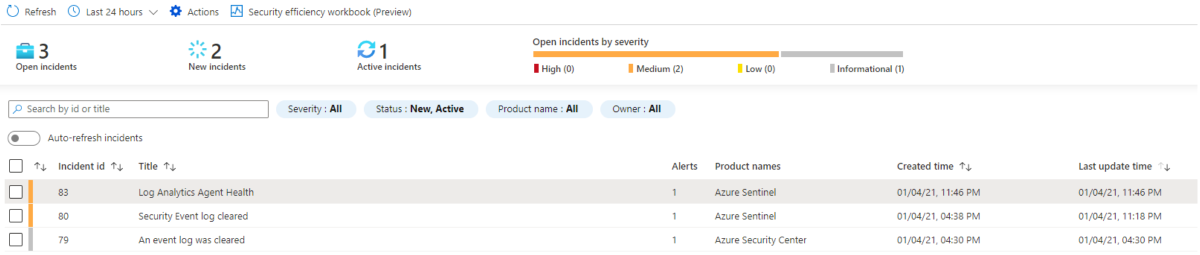
After all these steps, you should be ready to automate Azure Sentinel with PowerShell.

Part 1 – Incident Management using PowerShell

Get a specific incident

Summary

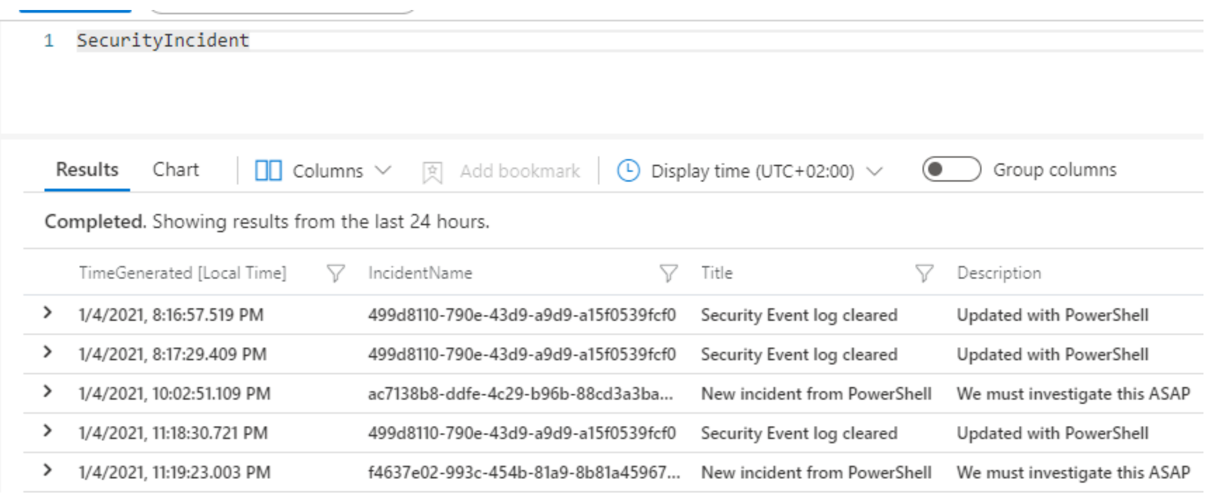
Most of the code examples include the **$AzureSentinelWorkSpaceInfo** variable. That's our hash table where we have stored our **resource group name** and **Log Analytics workspace name**. In the below code example, we are querying only one specific incident. As you see from the code block that we need to specify the **IncidentID** parameter. By default, the Azure Sentinel portal doesn't show that information, and you need to query that from the **SecurityIncident** table.



Severity : **All** | Status : **New, Active** | Product name : **All** | Owner : **All**

□	Incident id	Title	Alerts	Product names	Created time	Last update time
□	83	Log Analytics Agent Health	1	Azure Sentinel	01/04/21, 11:46 PM	01/04/21, 11:46 PM
□	80	Security Event log cleared	1	Azure Sentinel	01/04/21, 04:38 PM	01/04/21, 11:18 PM
□	79	An event log was cleared	1	Azure Security Center	01/04/21, 04:30 PM	01/04/21, 04:30 PM

Azure Sentinel portal



```
SecurityIncident
```

Completed. Showing results from the last 24 hours.

TimeGenerated [Local Time]	IncidentName	Title	Description
1/4/2021, 8:16:57.519 PM	499d8110-790e-43d9-a9d9-a15f0539fcf0	Security Event log cleared	Updated with PowerShell
1/4/2021, 8:17:29.409 PM	499d8110-790e-43d9-a9d9-a15f0539fcf0	Security Event log cleared	Updated with PowerShell
1/4/2021, 10:02:51.109 PM	ac7138b8-ddfe-4c29-b96b-88cd3a3ba...	New incident from PowerShell	We must investigate this ASAP
1/4/2021, 11:18:30.721 PM	499d8110-790e-43d9-a9d9-a15f0539fcf0	Security Event log cleared	Updated with PowerShell
1/4/2021, 11:19:23.003 PM	f4637e02-993c-454b-81a9-8b81a45967...	New incident from PowerShell	We must investigate this ASAP

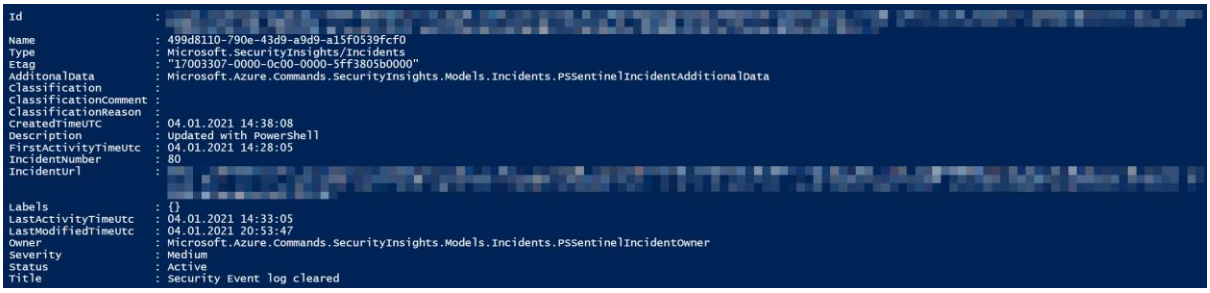
SecurityIncident table

Copy the value from the **IncidentName** column, and you should see the incident details with PowerShell.

Code example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$IncidentID = "499d8110-790e-43d9-a9d9-a15f0539fcf0"
Get-AzSentinelIncident @AzureSentinelWorkSpaceInfo -IncidentId $IncidentID
```

Output



```
Id                    : [redacted]

Name                  : 499d8110-790e-43d9-a9d9-a15f0539fcf0
Type                  : Microsoft.SecurityInsights/Incidents
Etag                  : "17003307-0000-0c00-0000-5ff3805b0000"
AdditonalData         : Microsoft.Azure.Commands.SecurityInsights.Models.Incidents.PSSentinelIncidentAdditionalData
Classification        :
ClassificationComment :
ClassificationReason  :
CreatedTimeUTC        : 04.01.2021 14:38:08
Description           : Updated with PowerShell
FirstActivityTimeUtc  : 04.01.2021 14:28:05
IncidentNumber        : 80
IncidentUrl           : [redacted]

Labels                : {}
LastActivityTimeUtc   : 04.01.2021 14:33:05
LastModifiedTimeUtc   : 04.01.2021 20:53:47
Owner                 : Microsoft.Azure.Commands.SecurityInsights.Models.Incidents.PSSentinelIncidentOwner
Severity              : Medium
Status                : Active
Title                 : Security Event log cleared
```

List all incidents

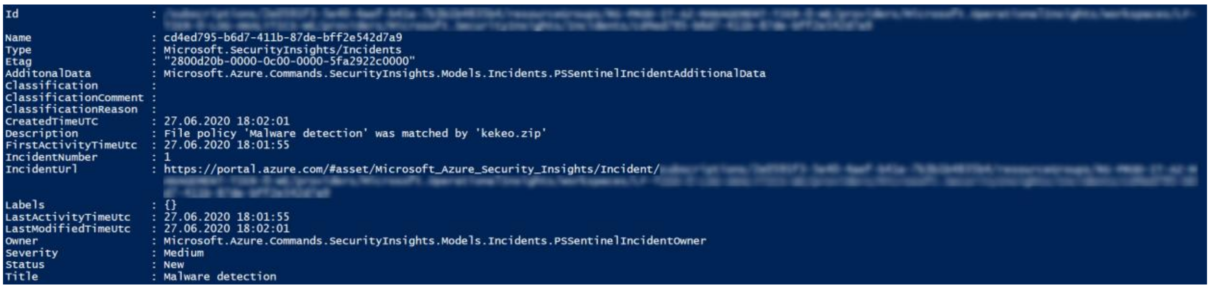
Summary

Get-AzSentinelIncident cmdlet allows you to query all the incidents. Just run the cmdlet with your environment information, and it should list all the incidents. If it is needed, you can do the filtering based on the **CreatedTimeUTC** property.

Code example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

Get-AzSentinelIncident @AzureSentinelWorkSpaceInfo
```

Output



```
Id                     : [redacted]

Name                   : cd4ed795-b6d7-411b-87de-bff2e542d7a9
Type                   : Microsoft.SecurityInsights/Incidents
Etag                   : "2800d20b-0000-0c00-0000-5fa2922c0000"
AdditonalData          : Microsoft.Azure.Commands.SecurityInsights.Models.Incidents.PSSentinelIncidentAdditionalData
Classification         :
ClassificationComment  :
ClassificationReason   :
CreatedTimeUTC         : 27.06.2020 18:02:01
Description            : File policy 'Malware detection' was matched by 'kekeo.zip'
FirstActivityTimeUtc   : 27.06.2020 18:01:55
IncidentNumber         : 1
IncidentUrl            : https://portal.azure.com/#asset/Microsoft_Azure_Security_Insights/Incident/[redacted]

Labels                 : {}
LastActivityTimeUtc    : 27.06.2020 18:01:55
LastModifiedTimeUtc    : 27.06.2020 18:02:01
Owner                  : Microsoft.Azure.Commands.SecurityInsights.Models.Incidents.PSSentinelIncidentOwner
Severity               : Medium
Status                 : New
Title                  : Malware detection
```

Get all incidents and order by CreatedTimeUTC property

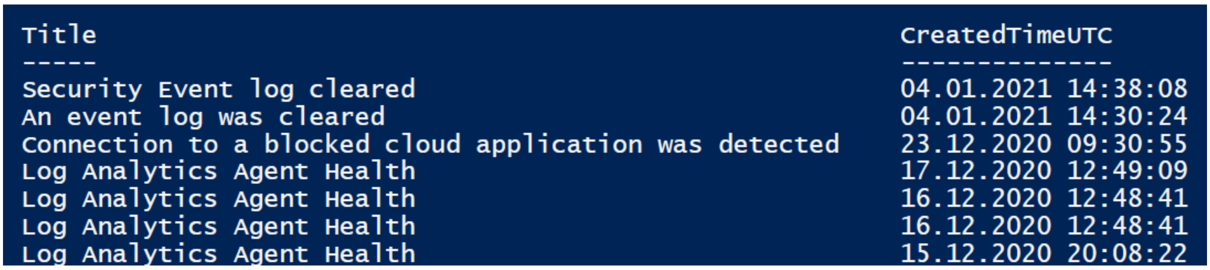
Summary

In this example, we have selected only two different properties using the **Select-Object** cmdlet – **Title** and **CreatedTimeUTC** and then sorting the results based on the **CreatedTimeUTC** property.

Code example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

Get-AzSentinelIncident @AzureSentinelWorkSpaceInfo |
    Select-Object -Property Title,CreatedTimeUTC |
        Sort-Object -Property CreatedTimeUTC -Descending
```

Output



```
Title                                                     CreatedTimeUTC
-----                                                     --------------
Security Event log cleared                                04.01.2021 14:38:08
An event log was cleared                                  04.01.2021 14:30:24
Connection to a blocked cloud application was detected    23.12.2020 09:30:55
Log Analytics Agent Health                                17.12.2020 12:49:09
Log Analytics Agent Health                                16.12.2020 12:48:41
Log Analytics Agent Health                                16.12.2020 12:48:41
Log Analytics Agent Health                                15.12.2020 20:08:22
```

Get all incidents and convert CreatedTimeUTC property to local DateTime

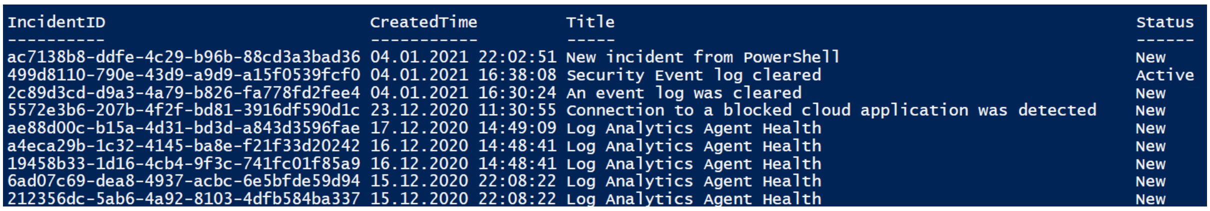
Summary

As you saw from the previous example, incident creation dates are in the UTC time zone. To convert the dates into the local time zone, we need to add one additional function. I'm not the author of that function, and it is taken from the ScriptingGuy blog.

Code example

```
Function Convert-UTCtoLocal
{

#Source - https://devblogs.microsoft.com/scripting/powertip-convert-from-utc-to-
my-local-time-zone/ PowerTip: Convert from UTC to my local time zone | Scripting
Blog (microsoft.com)
#Author - Thomas Rayner

    Param(
        [Parameter(Mandatory=$True)]
        [String]$UTCTime
        )

    $CurrentTimeZone = (Get-WmiObject win32_timezone).StandardName
    $TimeZone = [System.TimeZoneInfo]::FindSystemTimeZoneById($CurrentTimeZone)
    $LocalTime = [System.TimeZoneInfo]::ConvertTimeFromUtc($UTCTime, $TimeZone)

    $LocalTime
}

$ProcessedIncidents = @()

$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$Incidents = Get-AzSentinelIncident @AzureSentinelWorkSpaceInfo
foreach($Incident in $Incidents){

    $IncidentDetails = [ORDERED]@{
        IncidentID = $Incident.Name
        CreatedTime = Convert-UTCtoLocal -UTCTime $Incident.CreatedTimeUTC
        Title = $Incident.Title
        Status = $Incident.Status
    }

    $PoshObject = New-Object -TypeName PSObject -Property $IncidentDetails
    $ProcessedIncidents += $PoshObject
}
$ProcessedIncidents
```

Output



```
IncidentID                           CreatedTime         Title                                                     Status
----------                           -----------         -----                                                     ------
ac7138b8-ddfe-4c29-b96b-88cd3a3bad36 04.01.2021 22:02:51 New incident from PowerShell                              New
499d8110-790e-43d9-a9d9-a15f0539fcf0 04.01.2021 16:38:08 Security Event log cleared                                Active
2c89d3cd-d9a3-4a79-b826-fa778fd2fee4 04.01.2021 16:30:24 An event log was cleared                                  New
5572e3b6-207b-4f2f-bd81-3916df590d1c 23.12.2020 11:30:55 Connection to a blocked cloud application was detected    New
ae88d00c-b15a-4d31-bd3d-a843d3596fae 17.12.2020 14:49:09 Log Analytics Agent Health                                New
a4eca29b-1c32-4145-ba8e-f21f33d20242 16.12.2020 14:48:41 Log Analytics Agent Health                                New
19458b33-1d16-4cb4-9f3c-741fc01f85a9 16.12.2020 14:48:41 Log Analytics Agent Health                                New
6ad07c69-dea8-4937-acbc-6e5bfde59d94 15.12.2020 22:08:22 Log Analytics Agent Health                                New
212356dc-5ab6-4a92-8103-4dfb584ba337 15.12.2020 22:08:22 Log Analytics Agent Health                                New
```

Update incident details

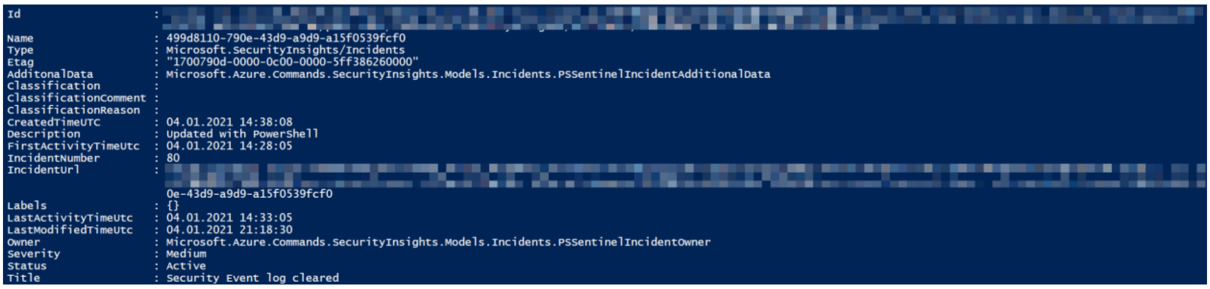
Summary

Changing the incident owner requires us to install the **Azure AD PowerShell** module. You can take the incident owner information manually from the Azure AD portal too, but most likely, it would be easier to use Azure AD PowerShell cmdlets for that. Run the **Get-AzureADUser** cmdlet and get the user details. After that, you can use the **New-AzSentinelIncidentOwner** cmdlet to create the owner object. Finally, run the **Update-AzSentinelIncident** command.

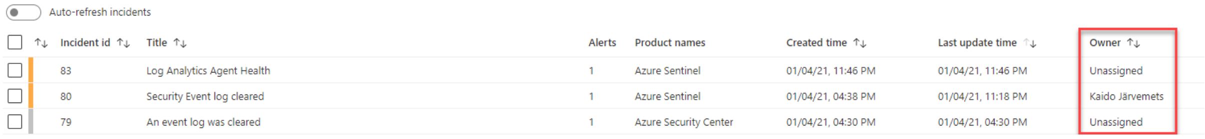
Code example

```
Connect-AzureAD

$AzureADUserDetails = Get-AzureADUser -ObjectId "John@Contoso.com"
$IncidentID = "499d8110-790e-43d9-a9d9-a15f0539fcf0"

$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$IncidentOwnerDetails = @{
        AssignedTo = $AzureADUserDetails.DisplayName
        Email = $AzureADUserDetails.Mail
        ObjectId = $AzureADUserDetails.ObjectId
        UserPrincipalName = $AzureADUserDetails.UserPrincipalName
}

$IncidentOwner = New-AzSentinelIncidentOwner @IncidentOwnerDetails

Update-AzSentinelIncident @AzureSentinelWorkSpaceInfo -IncidentID $IncidentID -
Owner $IncidentOwner -Status Active
```

Output



```
Id                     : [redacted]

Name                   : 499d8110-790e-43d9-a9d9-a15f0539fcf0
Type                   : Microsoft.SecurityInsights/Incidents
Etag                   : "1700790d-0000-0c00-0000-5ff386260000"
AdditonalData          : Microsoft.Azure.Commands.SecurityInsights.Models.Incidents.PSSentinelIncidentAdditionalData
Classification         :
ClassificationComment  :
ClassificationReason   :
CreatedTimeUTC         : 04.01.2021 14:38:08
Description            : Updated with PowerShell
FirstActivityTimeUtc   : 04.01.2021 14:28:05
IncidentNumber         : 80
IncidentUrl            : [redacted]
                         0e-43d9-a9d9-a15f0539fcf0
Labels                 : {}
LastActivityTimeUtc    : 04.01.2021 14:33:05
LastModifiedTimeUtc    : 04.01.2021 21:18:30
Owner                  : Microsoft.Azure.Commands.SecurityInsights.Models.Incidents.PSSentinelIncidentOwner
Severity               : Medium
Status                 : Active
Title                  : Security Event log cleared
```



□	Incident id ↑↓	Title ↑↓	Alerts	Product names	Created time ↑↓	Last update time ↑↓	Owner ↑↓
□	83	Log Analytics Agent Health	1	Azure Sentinel	01/04/21, 11:46 PM	01/04/21, 11:46 PM	Unassigned
□	80	Security Event log cleared	1	Azure Sentinel	01/04/21, 04:38 PM	01/04/21, 11:18 PM	Kaido Järvemets
□	79	An event log was cleared	1	Azure Security Center	01/04/21, 04:30 PM	01/04/21, 04:30 PM	Unassigned

Updated incident owner

Add a comment to an incident

Summary

Azure Sentinel allows us to add HTML based comments too. You can add tables or just formatted texts. The first example uses HTML tags, and the second one is just a regular comment without any formatting.

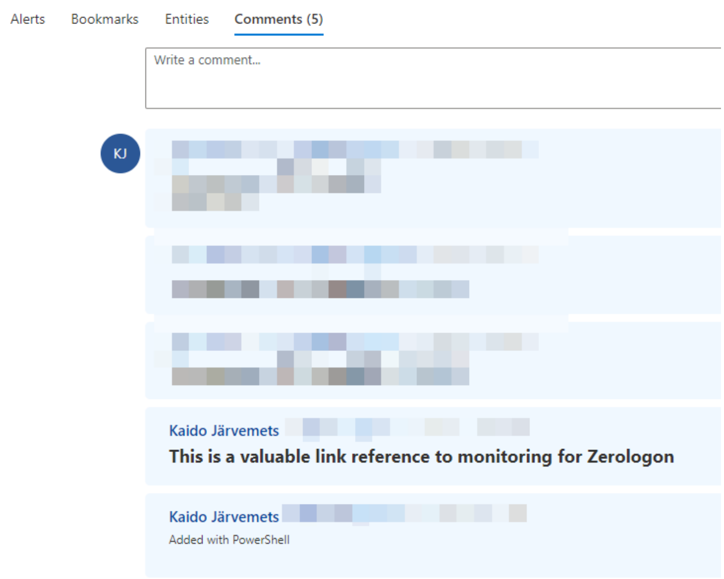
Code example 1

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$IncidentID = "499d8110-790e-43d9-a9d9-a15f0539fcf0"

New-AzSentinelIncidentComment @AzureSentinelWorkSpaceInfo -IncidentId $IncidentID
-Message "<h2>We can use HTML too!!!</h2>"
```

Code example 2

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$IncidentID = "499d8110-790e-43d9-a9d9-a15f0539fcf0"

New-AzSentinelIncidentComment @AzureSentinelWorkSpaceInfo -IncidentId $IncidentID
-Message "We need to investigate this ASAP"
```

Output



[redacted]

[redacted]

[redacted]

Kaido Järvemets [redacted]
This is a valuable link reference to monitoring for Zerologon

Kaido Järvemets [redacted]
Added with PowerShell

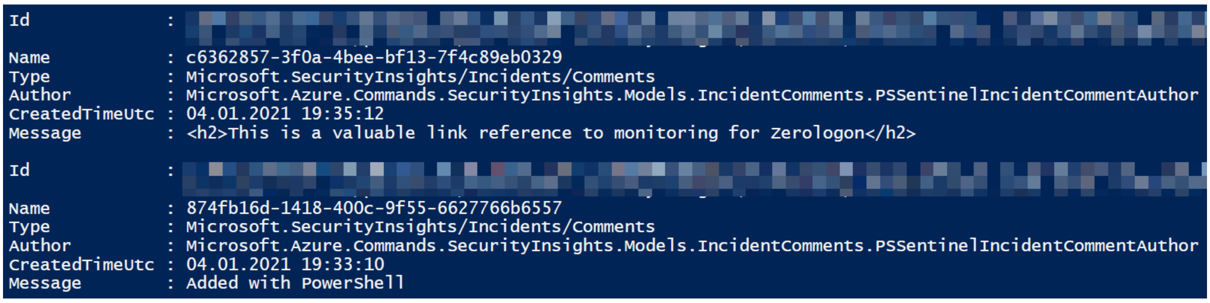
Read incident comments

Summary

Code example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$IncidentID = "499d8110-790e-43d9-a9d9-a15f0539fcf0"
Get-AzSentinelIncidentComment @AzureSentinelWorkSpaceInfo -IncidentId $IncidentID
```

Output



```
Id             : [redacted]

Name           : c6362857-3f0a-4bee-bf13-7f4c89eb0329
Type           : Microsoft.SecurityInsights/Incidents/Comments
Author         : Microsoft.Azure.Commands.SecurityInsights.Models.IncidentComments.PSSentinelIncidentCommentAuthor
CreatedTimeUtc : 04.01.2021 19:35:12
Message        : <h2>This is a valuable link reference to monitoring for Zerologon</h2>

Id             : [redacted]

Name           : 874fb16d-1418-400c-9f55-6627766b6557
Type           : Microsoft.SecurityInsights/Incidents/Comments
Author         : Microsoft.Azure.Commands.SecurityInsights.Models.IncidentComments.PSSentinelIncidentCommentAuthor
CreatedTimeUtc : 04.01.2021 19:33:10
Message        : Added with PowerShell
```

Create an incident

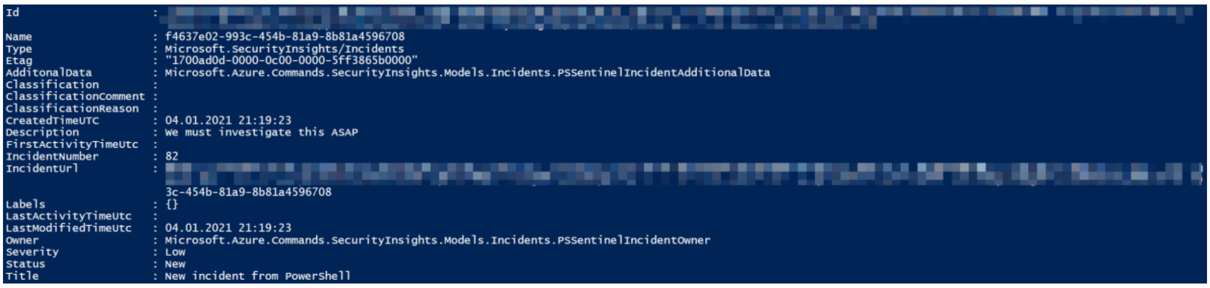
Summary

New-AzSentinelIncident cmdlet allows you to create new incidents. The strange thing is that the data source will be empty, and no investigation isn't available.

Code example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

New-AzSentinelIncident @AzureSentinelWorkSpaceInfo -Title "New incident from
PowerShell" -Description "We must investigate this ASAP" -Severity Low -Status
New
```

Output



```
Id                     : [redacted]

Name                   : f4637e02-993c-454b-81a9-8b81a4596708
Type                   : Microsoft.SecurityInsights/Incidents
Etag                   : "1700ad0d-0000-0c00-0000-5ff3865b0000"
AdditonalData          : Microsoft.Azure.Commands.SecurityInsights.Models.Incidents.PSSentinelIncidentAdditionalData
Classification         :
ClassificationComment  :
ClassificationReason   :
CreatedTimeUTC         : 04.01.2021 21:19:23
Description            : We must investigate this ASAP
FirstActivityTimeUtc   :
IncidentNumber         : 82
IncidentUrl            : [redacted]

                         3c-454b-81a9-8b81a4596708
Labels                 : {}
LastActivityTimeUtc    :
LastModifiedTimeUtc    : 04.01.2021 21:19:23
Owner                  : Microsoft.Azure.Commands.SecurityInsights.Models.Incidents.PSSentinelIncidentOwner
Severity               : Low
Status                 : New
Title                  : New incident from PowerShell
```

Remove incident

Summary

Remove-AzSentinelIncident removes the incident without any confirmations.

Code example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$IncidentID = "499d8110-790e-43d9-a9d9-a15f0539fcf0"
Remove-AzSentinelIncident @AzureSentinelWorkSpaceInfo -IncidentId $IncidentID
```

Output

The **Remove-AzSentinelIncident** cmdlet should return "**success**" if the removal was successful.

Part 2 – Alert Rule Management using PowerShell

Get all enabled Analytics rules

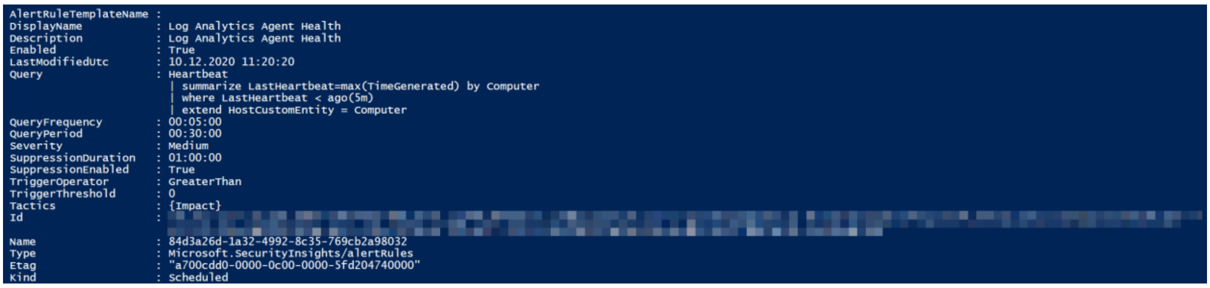
Summary

Get-AzSentinelAlertRule cmdlet lists all the enabled Analytics rules.

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

Get-AzSentinelAlertRule @AzureSentinelWorkSpaceInfo
```

Output



```
AlertRuleTemplateName :
DisplayName           : Log Analytics Agent Health
Description           : Log Analytics Agent Health
Enabled               : True
LastModifiedUtc       : 10.12.2020 11:20:20
Query                 : Heartbeat
                        | summarize LastHeartbeat=max(TimeGenerated) by Computer
                        | where LastHeartbeat < ago(5m)
                        | extend HostCustomEntity = Computer
QueryFrequency        : 00:05:00
QueryPeriod           : 00:30:00
Severity              : Medium
SuppressionDuration   : 01:00:00
SuppressionEnabled    : True
TriggerOperator       : GreaterThan
TriggerThreshold      : 0
Tactics               : {Impact}
Id                    : [redacted]

Name                  : 84d3a26d-1a32-4992-8c35-769cb2a98032
Type                  : Microsoft.SecurityInsights/alertRules
Etag                  : "a700cdd0-0000-0c00-0000-5fd204740000"
Kind                  : Scheduled
```

Get Analytics rule action

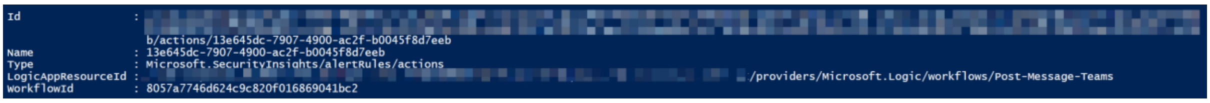
Summary

Azure Sentinel allows you to configure automated response actions to your analytics rules. **Get-AzSentinelAlertRuleAction** lists the configured playbooks. Use the **Get-AzSentinelAlertRule** cmdlet to get the **AlertRuleID** parameter value. Check the **Name** property.

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$AlertRuleId = "84d3a26d-1a32-4992-8c35-769cb2a98032"
Get-AzSentinelAlertRuleAction @AzureSentinelWorkSpaceInfo -AlertRuleId
$AlertRuleId
```

Output



```
Id                 : [redacted]
                     b/actions/13e645dc-7907-4900-ac2f-b0045f8d7eeb
Name               : 13e645dc-7907-4900-ac2f-b0045f8d7eeb
Type               : Microsoft.SecurityInsights/alertRules/actions
LogicAppResourceId : [redacted]/providers/Microsoft.Logic/workflows/Post-Message-Teams
WorkflowId         : 8057a7746d624c9c820f016869041bc2
```

Get Analytics rule action detailed information

Summary

In the previous example, we queried the configured playbook. Still, if you want more information about the configured playbook, we need to execute the **Get-AzLogicApp** cmdlet. In the below code example, I'm also using the **Split-Path** cmdlet. That gives me the configured playbook name.

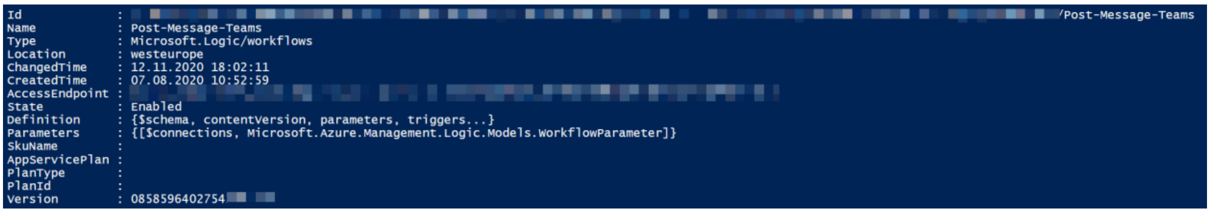
If you have multiple playbooks configured under the **Analytics rule**, you need to change the code slightly. Currently, the example assumes that you have only one playbook per the **Analytics rule**.

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$LogicAppsInfo = @{
    ResourceGroupName = "RG-PROD-IT-LOGIC-APPS-WE"
}

$AlertRuleId = "84d3a26d-1a32-4992-8c35-769cb2a98032"
$AlertRuleAction = Get-AzSentinelAlertRuleAction @AzureSentinelWorkSpaceInfo -
AlertRuleId $AlertRuleId

$AlertRuleActionName = $AlertRuleAction.LogicAppResourceId | Split-Path -Leaf
Get-AzLogicApp @LogicAppsInfo -Name $AlertRuleActionName
```

Output

You should see the following information:



```
Id             : [redacted]/Post-Message-Teams
Name           : Post-Message-Teams
Type           : Microsoft.Logic/workflows
Location       : westeurope
ChangedTime    : 12.11.2020 18:02:11
CreatedTime    : 07.08.2020 10:52:59
AccessEndpoint : [redacted]
State          : Enabled
Definition     : {$schema, contentVersion, parameters, triggers...}
Parameters     : {[$connections, Microsoft.Azure.Management.Logic.Models.WorkflowParameter]}
SkuName        :
AppServicePlan :
PlanType       :
PlanId         :
Version        : 0858596402754[redacted]
```

List all Analytics rule templates

Summary

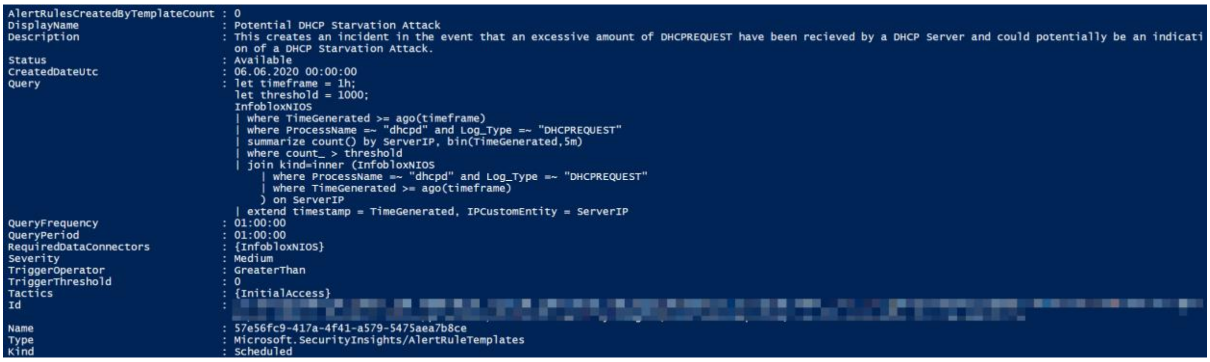
Get-AzSentinelAlertRuleTemplate lists all the available Analytics rule templates.

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

Get-AzSentinelAlertRuleTemplate @AzureSentinelWorkSpaceInfo
```

Output

You should see the following information:



```
AlertRulesCreatedByTemplateCount : 0
DisplayName                      : Potential DHCP Starvation Attack
Description                      : This creates an incident in the event that an excessive amount of DHCPREQUEST have been recieved by a DHCP Server and could potentially be an indicati
                                   on of a DHCP Starvation Attack.
Status                           : Available
CreatedDateUtc                   : 06.06.2020 00:00:00
Query                            : let timeframe = 1h;
                                   let threshold = 1000;
                                   InfobloxNIOS
                                   | where TimeGenerated >= ago(timeframe)
                                   | where ProcessName =~ "dhcpd" and Log_Type =~ "DHCPREQUEST"
                                   | summarize count() by ServerIP, bin(TimeGenerated,5m)
                                   | where count_ > threshold
                                   | join kind=inner (InfobloxNIOS
                                       | where ProcessName =~ "dhcpd" and Log_Type =~ "DHCPREQUEST"
                                       | where TimeGenerated >= ago(timeframe)
                                      ) on ServerIP
                                   | extend timestamp = TimeGenerated, IPCustomEntity = ServerIP
QueryFrequency                   : 01:00:00
QueryPeriod                      : 01:00:00
RequiredDataConnectors           : {InfobloxNIOS}
Severity                         : Medium
TriggerOperator                  : GreaterThan
TriggerThreshold                 : 0
Tactics                          : {InitialAccess}
Id                               : [redacted]

Name                             : 57e56fc9-417a-4f41-a579-5475aea7b8ce
Type                             : Microsoft.SecurityInsights/AlertRuleTemplates
Kind                             : Scheduled
```

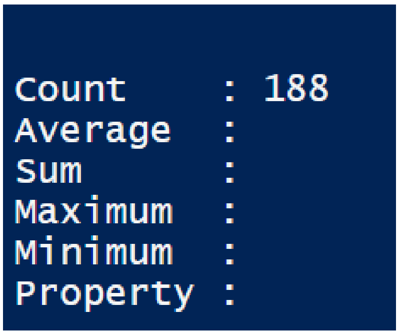
Count all the Analytics rule templates

Summary

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

Get-AzSentinelAlertRuleTemplate @AzureSentinelWorkSpaceInfo | Measure-Object
```

Output



```
Count    : 188
Average  :
Sum      :
Maximum  :
Minimum  :
Property :
```

List all Analytics rules and sort rules based on the Severity

Summary

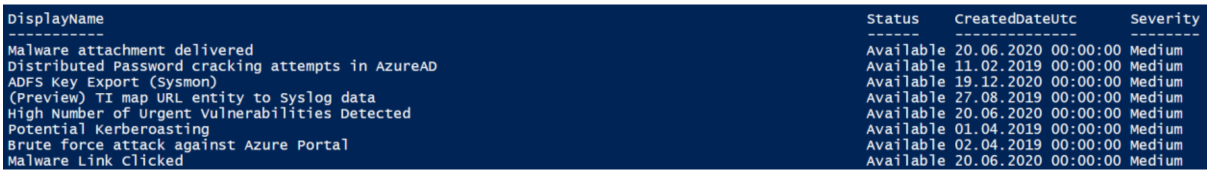
In this example, we have selected out only four properties - **DisplayName**, **Status**, **CreatedDateUtc**, and **Severity**. Then we are sorting the results based on the **Severity** property.

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

Get-AzSentinelAlertRuleTemplate @AzureSentinelWorkSpaceInfo |
    Select-Object -Property DisplayName,Status,CreatedDateUtc,Severity |
        Sort-Object -Property Severity -Descending
```

Output

The above code block should give you the following output:



```
DisplayName                                                                                       Status    CreatedDateUtc      Severity
-----------                                                                                       ------    --------------      --------
Malware attachment delivered                                                                      Available 20.06.2020 00:00:00 Medium
Distributed Password cracking attempts in AzureAD                                                 Available 11.02.2019 00:00:00 Medium
ADFS Key Export (Sysmon)                                                                          Available 19.12.2020 00:00:00 Medium
(Preview) TI map URL entity to Syslog data                                                        Available 27.08.2019 00:00:00 Medium
High Number of Urgent Vulnerabilities Detected                                                    Available 20.06.2020 00:00:00 Medium
Potential Kerberoasting                                                                           Available 01.04.2019 00:00:00 Medium
Brute force attack against Azure Portal                                                           Available 02.04.2019 00:00:00 Medium
Malware Link Clicked                                                                              Available 20.06.2020 00:00:00 Medium
```

List all Analytics rules and group by Severity

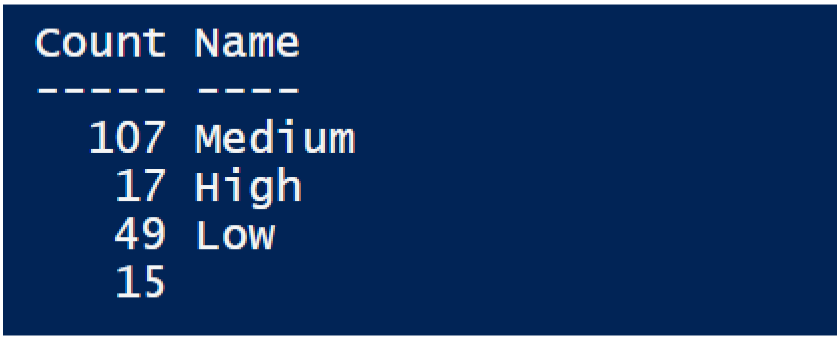
Summary

This code example counts different rule types based on the Severity property. Interestingly, we have 15 rules without any **Severity**.

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

Get-AzSentinelAlertRuleTemplate @AzureSentinelWorkSpaceInfo |
        Group-Object -Property Severity
```

Output



```
Count Name
----- ----
  107 Medium
   17 High
   49 Low
   15
```

List all Analytics rules where Data Sources contains "SecurityEvents"

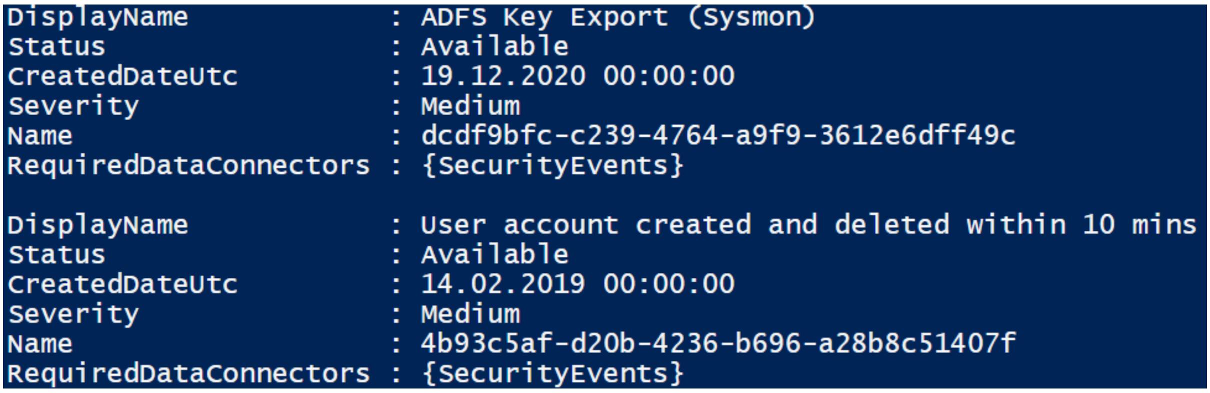
Summary

The following code example lists all the Analytics rules, where the **Data Source** contains "**SecurityEvents**". This example may be really handy when we are going to combine it with **Update-AzSentinelAlertRule** or **Update-AzSentinelAlertRuleAction** cmdlet. It allows us to filter out specific Analytics rules, and then we can enable all of them at once.

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

Get-AzSentinelAlertRuleTemplate @AzureSentinelWorkSpaceInfo |
    Where-Object {$PSItem.RequiredDataConnectors.ConnectorId -contains
"SecurityEvents"} |
        Select-Object -Property DisplayName,Status,CreatedDateUtc,Severity,Name
,RequiredDataConnectors |
            Sort-Object -Property Severity
```

Output



```
DisplayName            : ADFS Key Export (Sysmon)
Status                 : Available
CreatedDateUtc         : 19.12.2020 00:00:00
Severity               : Medium
Name                   : dcdf9bfc-c239-4764-a9f9-3612e6dff49c
RequiredDataConnectors : {SecurityEvents}

DisplayName            : User account created and deleted within 10 mins
Status                 : Available
CreatedDateUtc         : 14.02.2019 00:00:00
Severity               : Medium
Name                   : 4b93c5af-d20b-4236-b696-a28b8c51407f
RequiredDataConnectors : {SecurityEvents}
```

Filter Analytics rules based on the CreatedDateUtc property

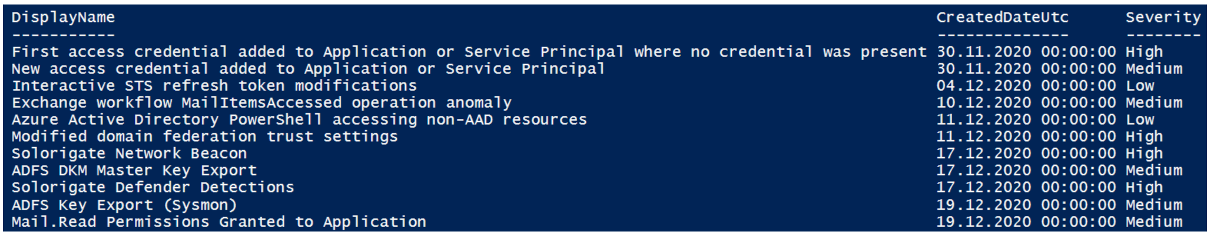
Summary

The good thing about Azure Sentinel is that Microsoft keeps adding new Analytics rules. This query prints out all the rules that have been added in the last 60 days.

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$TimeRange = (Get-Date).AddDays(-60)

$TimeRange = (Get-Date).AddDays(-60)
Get-AzSentinelAlertRuleTemplate @AzureSentinelWorkSpaceInfo |
    Where-Object {$PSItem.CreatedDateUtc -ge $TimeRange} |
        Select-Object -Property DisplayName,CreatedDateUtc,Severity |
            Sort-Object -Property CreatedDateUtc
```

Output



```
DisplayName                                                                                       CreatedDateUtc      Severity
-----------                                                                                       --------------      --------
First access credential added to Application or Service Principal where no credential was present 30.11.2020 00:00:00 High
New access credential added to Application or Service Principal                                   30.11.2020 00:00:00 Medium
Interactive STS refresh token modifications                                                       04.12.2020 00:00:00 Low
Exchange workflow MailItemsAccessed operation anomaly                                             10.12.2020 00:00:00 Medium
Azure Active Directory PowerShell accessing non-AAD resources                                     11.12.2020 00:00:00 Low
Modified domain federation trust settings                                                         11.12.2020 00:00:00 High
Solorigate Network Beacon                                                                         17.12.2020 00:00:00 High
ADFS DKM Master Key Export                                                                        17.12.2020 00:00:00 Medium
Solorigate Defender Detections                                                                    17.12.2020 00:00:00 High
ADFS Key Export (Sysmon)                                                                          19.12.2020 00:00:00 Medium
Mail.Read Permissions Granted to Application                                                      19.12.2020 00:00:00 Medium
```

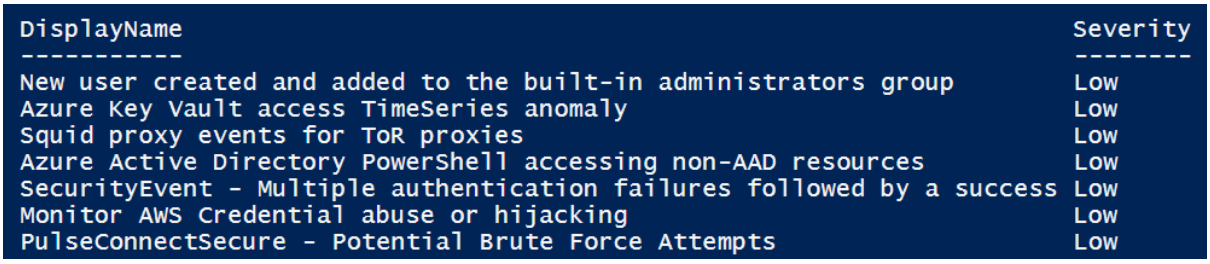
List all Low Severity based Analytics rules

Summary

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

Get-AzSentinelAlertRuleTemplate @AzureSentinelWorkSpaceInfo |
    Where-Object {$PSItem.Severity -eq "Low"} |
        Select-Object -Property DisplayName,Severity
```

Output



```
DisplayName                                                           Severity
-----------                                                           --------
New user created and added to the built-in administrators group       Low
Azure Key Vault access TimeSeries anomaly                             Low
Squid proxy events for ToR proxies                                    Low
Azure Active Directory PowerShell accessing non-AAD resources          Low
SecurityEvent - Multiple authentication failures followed by a success Low
Monitor AWS Credential abuse or hijacking                              Low
PulseConnectSecure - Potential Brute Force Attempts                    Low
```

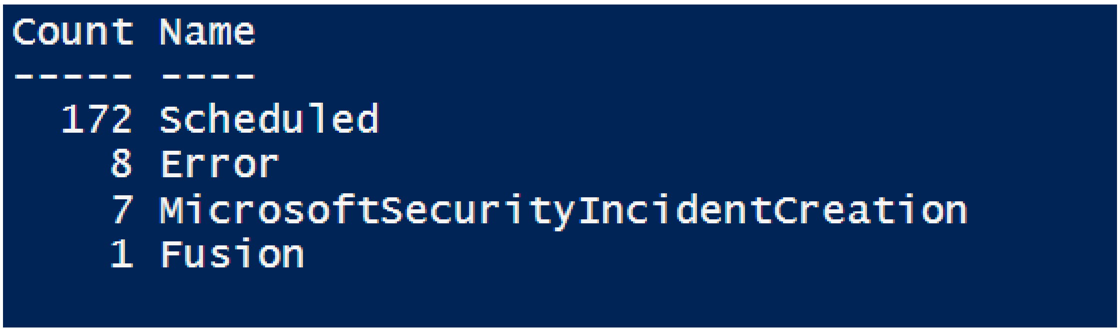
Count Analytics rule template types

Summary

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

Get-AzSentinelAlertRuleTemplate @AzureSentinelWorkSpaceInfo |
    Group-Object -Property Kind |
        Select-Object -Property Count,Name
```

Output



```
Count Name
----- ----
  172 Scheduled
    8 Error
    7 MicrosoftSecurityIncidentCreation
    1 Fusion
```

Create a new custom Analytics rule

Summary

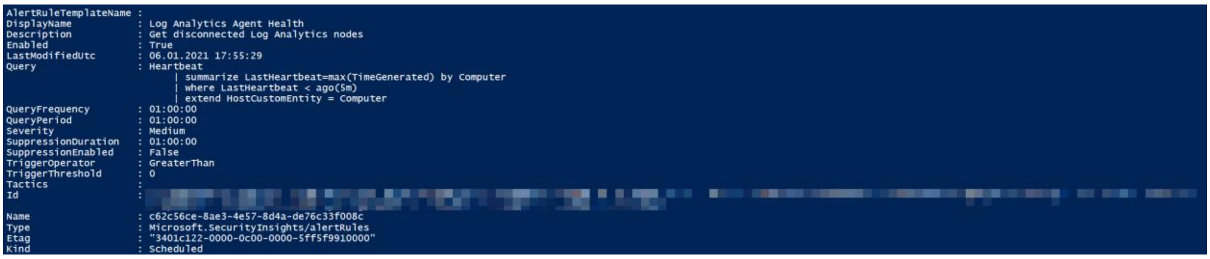
The **New-AzSentinelAlertRule** cmdlet creates a new Analytics rule. This example creates a new "**Scheduled**" based Analytics rule. If you have your own custom rules, then it would be much easier for you to import new rules.

Please remember that this is just a sample Analytics rule, and do not use it in production!

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$NewAnalyticsRuleData = @{
    Scheduled = $True
    Enabled = $True
    Query = "Heartbeat
    | summarize LastHeartbeat=max(TimeGenerated) by Computer
    | where LastHeartbeat < ago(5m)
    | extend HostCustomEntity = Computer"

    DisplayName = "TEST - Log Analytics Agent Health"
    Description = "Get disconnected Log Analytics nodes"
    QueryPeriod = (New-TimeSpan -Hours 1)
    QueryFrequency = (New-TimeSpan -Hours 1)
    TriggerThreshold = 0
    TriggerOperator = "GreaterThan" #Equal, GreaterThan, LessThan, NotEqual
    Severity = "Medium" # Low, Medium, High
}

New-AzSentinelAlertRule @AzureSentinelWorkSpaceInfo @NewAnalyticsRuleData
```

Output



```
AlertRuleTemplateName :
DisplayName           : Log Analytics Agent Health
Description           : Get disconnected Log Analytics nodes
Enabled               : True
LastModifiedUtc       : 06.01.2021 17:55:29
Query                 : Heartbeat
                            | summarize LastHeartbeat=max(TimeGenerated) by Computer
                            | where LastHeartbeat < ago(5m)
                            | extend HostCustomEntity = Computer
QueryFrequency        : 01:00:00
QueryPeriod           : 01:00:00
Severity              : Medium
SuppressionDuration   : 01:00:00
SuppressionEnabled    : False
TriggerOperator       : GreaterThan
TriggerThreshold      : 0
Tactics               :
Id                    : [redacted]

Name                  : c62c56ce-8ae3-4e57-8d4a-de76c33f008c
Type                  : Microsoft.SecurityInsights/alertRules
Etag                  : "3401c122-0000-0c00-0000-5ff5f9910000"
Kind                  : Scheduled
```

Add a new automated response for the Analytics rule

Summary

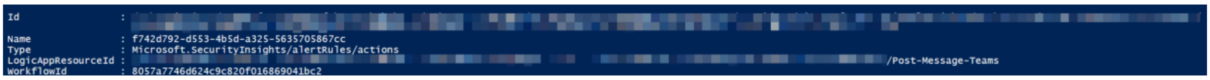
The **New-AzSentinelAlertRule** cmdlet does not allow us to add an automated response immediately, but we can use the **New-AzSentinelAlertRuleAction** cmdlet for that activity. Before that, we need to query our playbook information using the **Get-AzLogicApp** and **Get-AzLogicAppTriggerCallbackUrl** cmdlets. We can then pass that information to the **New-AzSentinelAlertRuleAction** cmdlet. Then, we should see the attached playbook under our Analytics rule.

In my case, all my Logic Apps are under one single resource group.

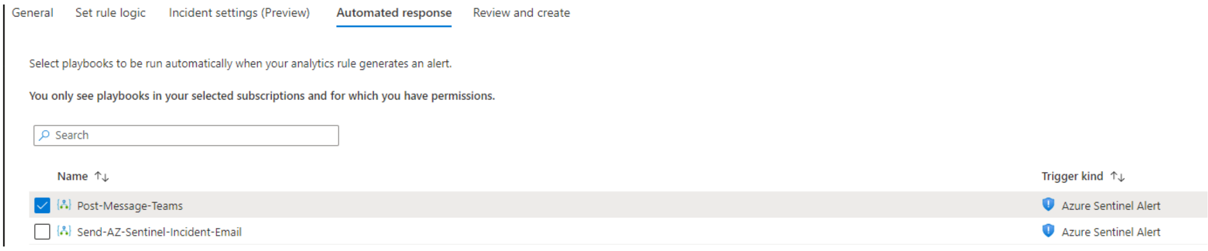
Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$LogicAppsInfo = @{
    ResourceGroupName = "RG-PROD-IT-LOGIC-APPS-WE"
    Name = "Post-Message-Teams"
}

$LogicAppResourceID = Get-AzLogicApp @LogicAppsInfo
$LogicAppTriggerURI = Get-AzLogicAppTriggerCallbackUrl @LogicAppsInfo -
TriggerName "When_a_response_to_an_Azure_Sentinel_alert_is_triggered"

$AnalyticsRule = Get-AzSentinelAlertRule @AzureSentinelWorkSpaceInfo |
    Where-Object {$PSItem.DisplayName -eq "Log Analytics Agent Health"}

New-AzSentinelAlertRuleAction @AzureSentinelWorkSpaceInfo -AlertRuleId
$AnalyticsRule.Name -LogicAppResourceId ($LogicAppResourceID.Id) -TriggerUri
($LogicAppTriggerURI.Value)
```

Output



```
Id                 : [redacted]
Name               : f742d792-d553-4b5d-a325-5635705867cc
Type               : Microsoft.SecurityInsights/alertRules/actions
LogicAppResourceId : [redacted]/Post-Message-Teams
WorkflowId         : 8057a7746d624c9c820f016869041bc2
```



Select playbooks to be run automatically when your analytics rule generates an alert.

You only see playbooks in your selected subscriptions and for which you have permissions.

	Name ↑↓	Trigger kind ↑↓
☑	Post-Message-Teams	Azure Sentinel Alert
☐	Send-AZ-Sentinel-Incident-Email	Azure Sentinel Alert

Configured playbook under the Analytics rule

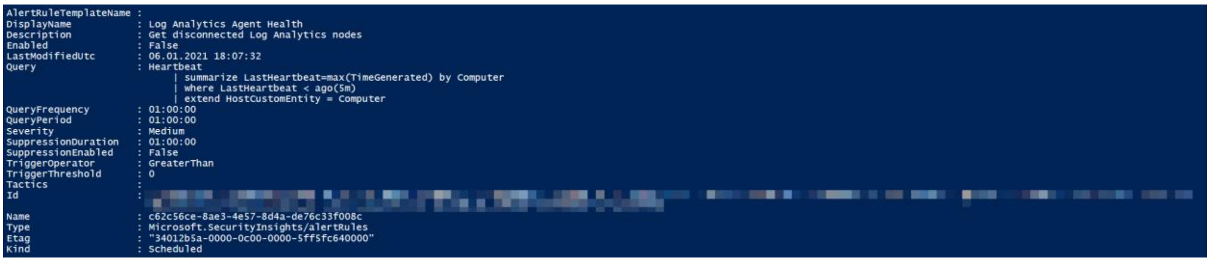
Disable enabled Analytics rule

Summary

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$AnalyticsRule = Get-AzSentinelAlertRule @AzureSentinelWorkSpaceInfo |
    Where-Object {$PSItem.DisplayName -eq "Log Analytics Agent Health"}

Update-AzSentinelAlertRule @AzureSentinelWorkSpaceInfo -AlertRuleId
$AnalyticsRule.Name -Disabled
```

Output



```
AlertRuleTemplateName :
DisplayName           : Log Analytics Agent Health
Description           : Get disconnected Log Analytics nodes
Enabled               : False
LastModifiedUtc       : 06.01.2021 18:07:32
Query                 : Heartbeat
                            | summarize LastHeartbeat=max(TimeGenerated) by Computer
                            | where LastHeartbeat < ago(5m)
                            | extend HostCustomEntity = Computer
QueryFrequency        : 01:00:00
QueryPeriod           : 01:00:00
Severity              : Medium
SuppressionDuration   : 01:00:00
SuppressionEnabled    : False
TriggerOperator       : GreaterThan
TriggerThreshold      : 0
Tactics               :
Id                    : [redacted]

Name                  : c62c56ce-8ae3-4e57-8d4a-de76c33f008c
Type                  : Microsoft.SecurityInsights/alertRules
Etag                  : "34012b5a-0000-0c00-0000-5ff5fc640000"
Kind                  : Scheduled
```

Remove automated response from the Analytics rule

Summary

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$AnalyticsRule = Get-AzSentinelAlertRule @AzureSentinelWorkSpaceInfo |
    Where-Object {$PSItem.DisplayName -eq "Log Analytics Agent Health"}

$AlertRuleAction = Get-AzSentinelAlertRuleAction @AzureSentinelWorkSpaceInfo -
AlertRuleId $AnalyticsRule.Name

Remove-AzSentinelAlertRuleAction @AzureSentinelWorkSpaceInfo -AlertRuleId
$AnalyticsRule.Name -ActionId $AlertRuleAction.Name
```

Output

The **Remove-AzSentinelAlertRuleAction** cmdlet should return "**success**" if the removal was successful.

Part 3 – Bookmark Management using PowerShell

Add new Bookmark

Summary

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$BookMarkQuery = @"
let AllWindowsServers =
Heartbeat
| where OSType == 'Windows' and OSType != "Linux"
| summarize arg_max(TimeGenerated, *) by SourceComputerId
| summarize makeset(Computer);
ProtectionStatus
| where Computer in (AllWindowsServers)
| sort by TimeGenerated desc
| summarize arg_max(TimeGenerated, *) by SourceComputerId
| summarize  count() by TypeofProtection, AMProductVersion
"@

$DisplayName = "Get Windows Defender Status from Windows Servers"
$Notes = "Please review"

New-AzSentinelBookmark @AzureSentinelWorkSpaceInfo -DisplayName $DisplayName -
Query $BookMarkQuery -Note $Notes
```

Output

Get Bookmarks

Summary

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

Get-AzSentinelBookmark @AzureSentinelWorkSpaceInfo
```

Output

Update Bookmark information

Summary

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$BookMark = Get-AzSentinelBookmark @AzureSentinelWorkSpaceInfo |
    Where-Object {$PSItem.DisplayName -eq "Get Windows Defender Status from
Windows Servers"}

$Notes = "Check out the Server1. Something seems wrong with that"
Update-AzSentinelBookmark @AzureSentinelWorkSpaceInfo -BookmarkId $BookMark.Name
-Note $Notes
```

Output

Remove Bookmark

Summary

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}

$BookMark = Get-AzSentinelBookmark @AzureSentinelWorkSpaceInfo |
    Where-Object {$PSItem.DisplayName -eq "Get Windows Defender Status from
Windows Servers"}

Remove-AzSentinelBookmark @AzureSentinelWorkSpaceInfo -BookmarkId $BookMark.Name
```

Output

Part 4 – Data Connector Management using PowerShell

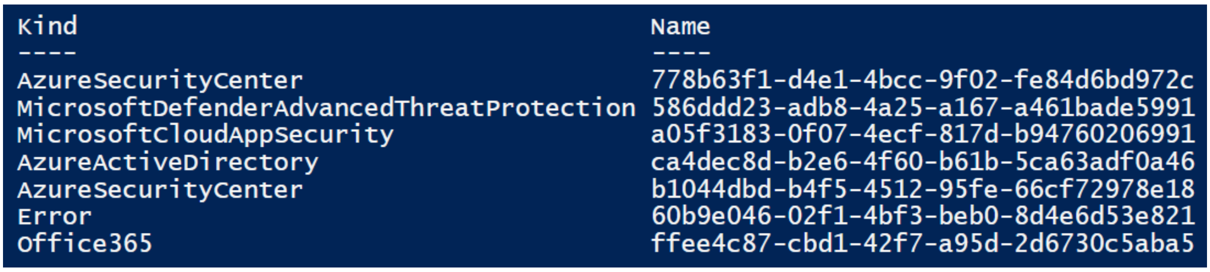
Get Data Connectors

Summary

Code Example

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}
Get-AzSentinelDataConnector @AzureSentinelWorkSpaceInfo |
    Select-Object -Property Kind,Name
```

Output



```
Kind                                      Name
----                                      ----
AzureSecurityCenter                       778b63f1-d4e1-4bcc-9f02-fe84d6bd972c
MicrosoftDefenderAdvancedThreatProtection 586ddd23-adb8-4a25-a167-a461bade5991
MicrosoftCloudAppSecurity                 a05f3183-0f07-4ecf-817d-b94760206991
AzureActiveDirectory                      ca4dec8d-b2e6-4f60-b61b-5ca63adf0a46
AzureSecurityCenter                       b1044dbd-b4f5-4512-95fe-66cf72978e18
Error                                     60b9e046-02f1-4bf3-beb0-8d4e6d53e821
Office365                                 ffee4c87-cbd1-42f7-a95d-2d6730c5aba5
```

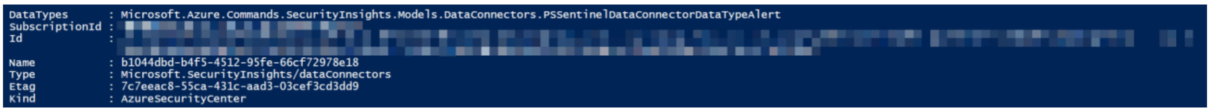
Configure Data Connectors

Summary

Code Example – Enable Azure Security Center

```
$AzureSentinelWorkSpaceInfo = @{
    ResourceGroupName = "RG-PROD-IT-AZ-MANAGEMENT-TIER-0-WE"
    WorkspaceName = "LF-TIER-0-LOG-ANALYTICS-WE"
}
New-AzSentinelDataConnector @AzureSentinelWorkSpaceInfo -AzureSecurityCenter -
SubscriptionId "%YOURSUBSCRIPTIONID%" -Alerts Enabled
```

Output



```
DataTypes      : Microsoft.Azure.Commands.SecurityInsights.Models.DataConnectors.PSSentinelDataConnectorDataTypeAlert
SubscriptionId : [redacted]
Id             : [redacted]

Name           : b1044dbd-b4f5-4512-95fe-66cf72978e18
Type           : Microsoft.SecurityInsights/dataConnectors
Etag           : 7c7eeac8-55ca-431c-aad3-03cef3cd3dd9
Kind           : AzureSecurityCenter
```