LAKEFOREST

Azure Sentinel incident management using PowerShell

Kaido Järvemets

Microsoft MVP: Enterprise Mobility, MCT, Security+

LAKEFOREST

Contents

Azure Sentinel incident management using PowerShell 1
Contents 2
Introduction 4
Part 1 – Incident Management using PowerShell 6
Get a specific incident 6
Summary 6
Code example 7
Output 7
List all incidents 8
Summary 8
Code example 8
Output 8
Get all incidents and order by CreatedTimeUTC property 9
Summary 9
Code example 9
Output 9
Get all incidents and convert CreatedTimeUTC property to local DateTime 10
Summary 10
Code example 10
Output 11
Update incident details 12
Summary 12
Code example 12
Output 12
Add a comment to an incident 13
Summary 13
Code example 1 13
Code example 2 13
Output 14

Read incident comments 15
Summary 15
Code example 15
Output 15
Create an incident 16
Summary 16
Code example 16
Output 16
Remove incident 17
Summary 17
Code example 17
Output 17

LAKEFOREST

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 incidents with the native cmdlets. Just to point out, that's the first version, and in version 0.1.0, we have these cmdlets around incident management:

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

As you see, not much but at least something to play with 😊

Here are the 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.

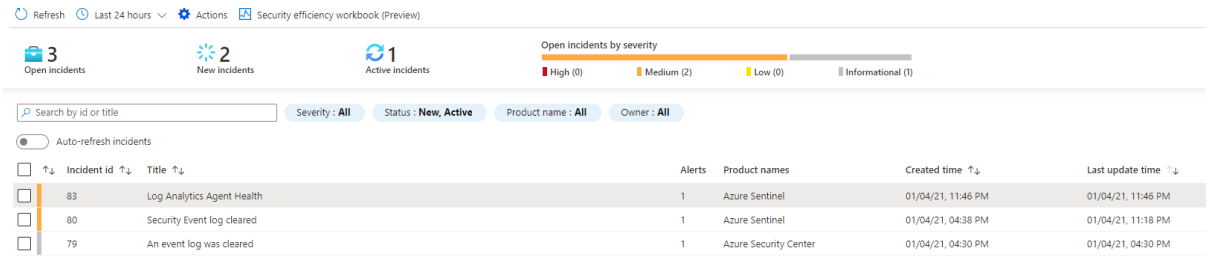
LAKEFOREST

Part 1 – Incident Management using PowerShell

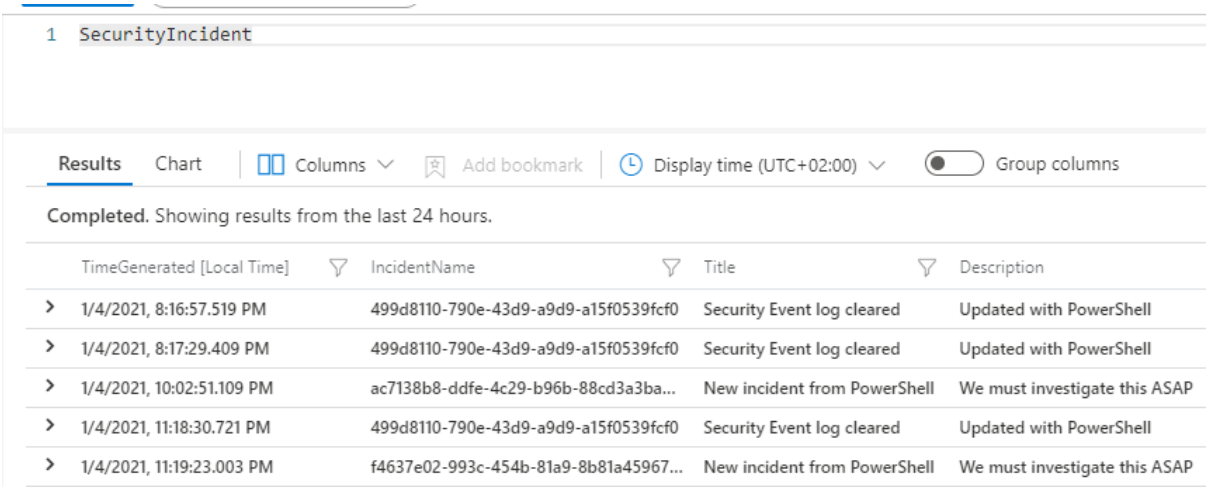
Get a specific incident

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.



Azure Sentinel portal



SecurityIncident table

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

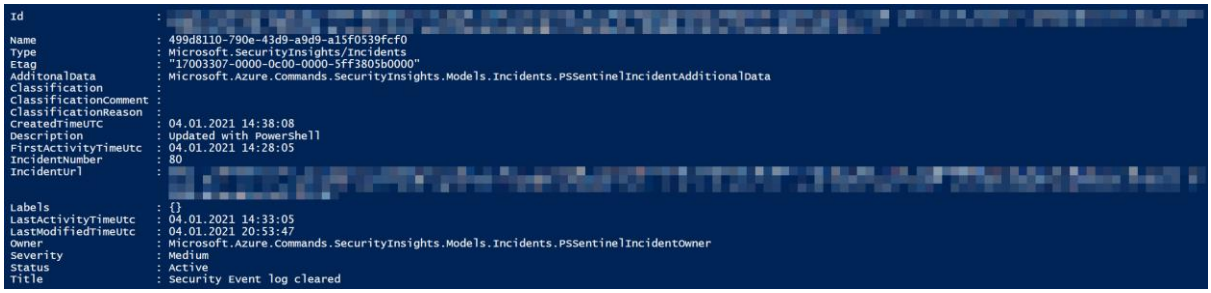
LAKEFOREST

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



LAKEFOREST

List all incidents

Summary

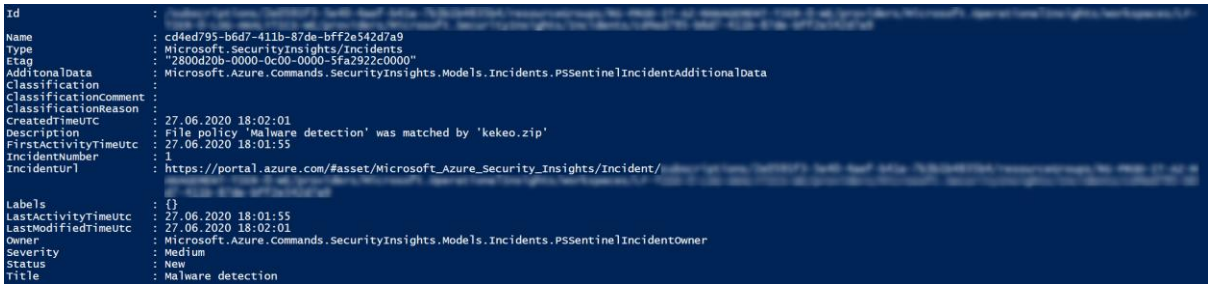
Get-AzSentinelIncident cmdlet allows you to query all the incidents. Just run the cmdlet with your environment information, and it should list all the incidents. If it is needed, you can do the filtering based on the **CreatedTimeUTC** property.

Code example

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

Get-AzSentinelIncident @AzureSentinelWorkSpaceInfo
```

Output



LAKEFOREST

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
```

LAKEFOREST

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
```

LAKEFOREST

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
```

LAKEFOREST

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
```

Auto-refresh incidents

	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

LAKEFOREST

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>"
```

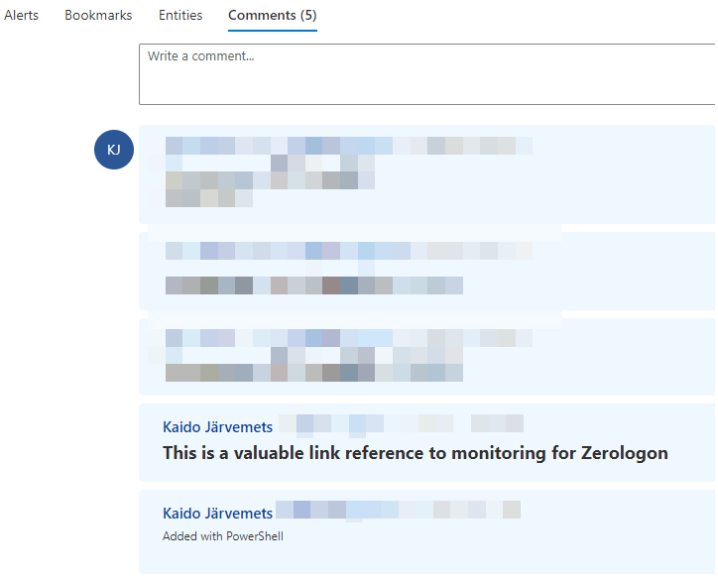
Code example 2

```
$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


Alerts
Bookmarks
Entities
Comments (5)
Write a comment...
KJ
Kaido Järvemets
This is a valuable link reference to monitoring for Zerologon
Kaido Järvemets
Added with PowerShell

LAKEFOREST

Read incident comments

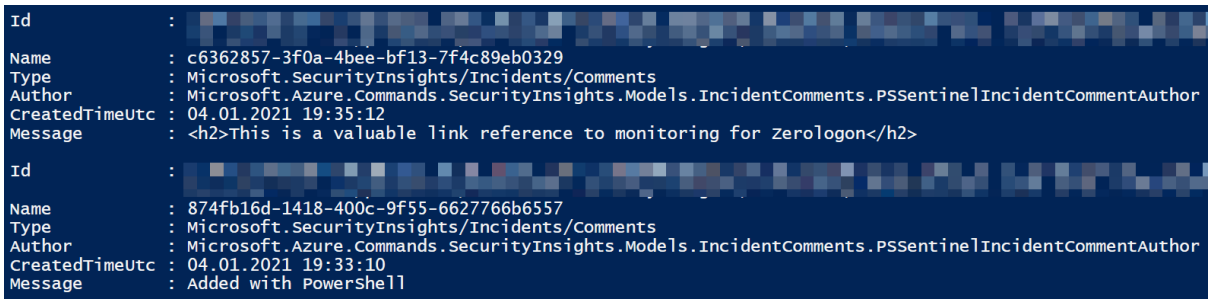
Summary

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



LAKEFOREST

Create an incident

Summary

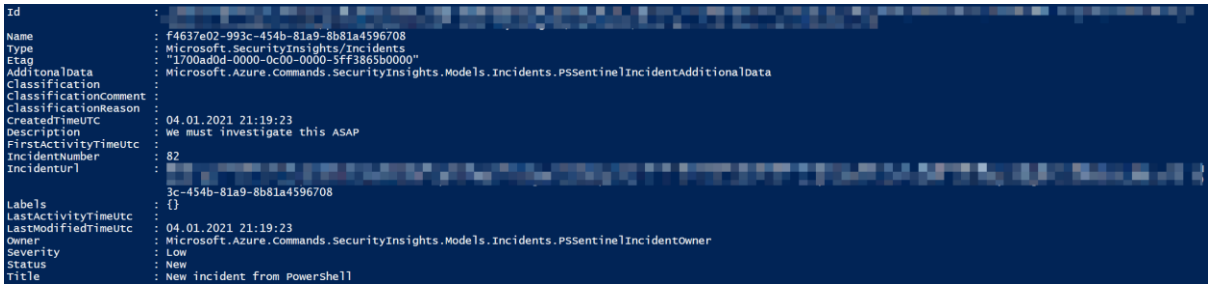
New-AzSentinelIncident cmdlet allows you to create new incidents. The strange thing is that the data source will be empty, and no investigation isn't available.

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



LAKEFOREST

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.