Azure Sentinel + Configuration Manager = Better together

Kaido Järvemets

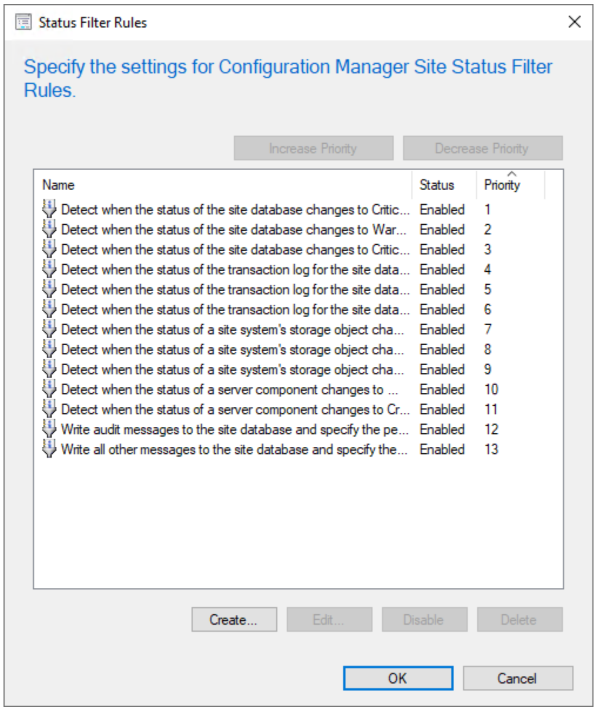
Microsoft MVP: Enterprise Mobility

me@kaidojarvemets.com

Description

A few months ago, Microsoft had an Azure Sentinel hackathon, and I thought that maybe I could build-up one POC solution for Azure Sentinel. My idea was to capture all the admin activities and send the data immediately off the Primary Site server. The reason why I decided to build this POC is that the built-in auditing isn't that good, and it needs some modernization. Luckily we have such good services like Log Analytics, Azure Sentinel, etc.

Configuration Manager allows you to create Status Filter Rules, and with that, you can save the activities to Event Log, but the data isn't that good. So that's why I decided to write extra PowerShell scripts to capture additional data for each Event.



Specify the settings for Configuration Manager Site Status Filter Rules.

Name	Status	Priority
Detect when the status of the site database changes to Critic...	Enabled	1
Detect when the status of the site database changes to War...	Enabled	2
Detect when the status of the site database changes to Critic...	Enabled	3
Detect when the status of the transaction log for the site data...	Enabled	4
Detect when the status of the transaction log for the site data...	Enabled	5
Detect when the status of the transaction log for the site data...	Enabled	6
Detect when the status of a site system's storage object cha...	Enabled	7
Detect when the status of a site system's storage object cha...	Enabled	8
Detect when the status of a site system's storage object cha...	Enabled	9
Detect when the status of a server component changes to ...	Enabled	10
Detect when the status of a server component changes to Cr...	Enabled	11
Write audit messages to the site database and specify the pe...	Enabled	12
Write all other messages to the site database and specify the...	Enabled	13

Configuration Manager Status Filter Rules

Currently included events in this POC

- Somebody added or removed a new administrator
- Configuration Manager Client Settings has been changed
- New Application deployment
- Change in Configuration Manager Site settings
- New Security Role
- Someone added a new Script
- Script execution against devices
- Someone changed a Script
- Script execution against a Collection

- Someone approved a Script
- Someone added a new Collection
- Collection has been changed
- Collection has been removed
- Someone changed a Task Sequence
- New Software Updates group
- Someone removed the Software Updates group
- Someone started a Remote Control session
- Remote Control session ended
- Someone removed a Device
- New Package

AZMEMCM Configuration

Pre-requisites

For this solution, you need:

- Configuration Manager Primary Site
  - Administrative permissions in Configuration Manager and underline OS
- Log Analytics Workspace
- Azure Sentinel (optional)
  - Azure Sentinel not required for this to work, but you can create different hunting queries

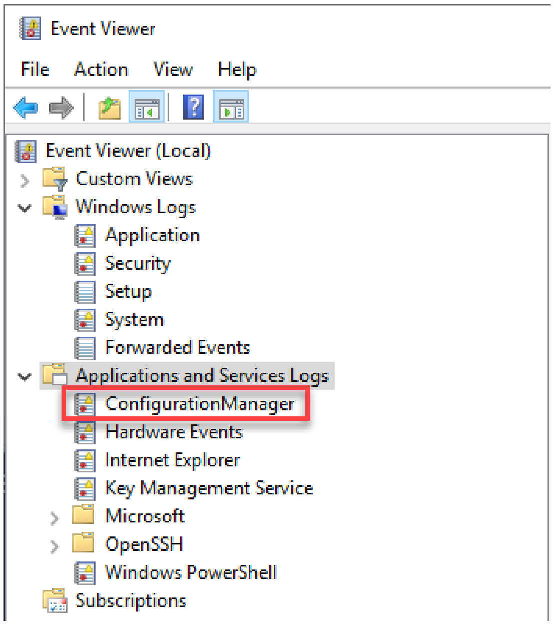
Configuration

Custom Event Log

<table>
<tr><td></td><td>On your Primary Site server, run PowerShell ISE as an administrator</td></tr>
<tr><td colspan="2">Run the following command to create custom Event Log for Configuration Manager:

```
New-EventLog -LogName "ConfigurationManager" -Source "ConfigurationManager"
```

After that, you should see the <b>ConfigurationManager</b> Event Log. We are going to store all the data there


</td></tr>
</table>

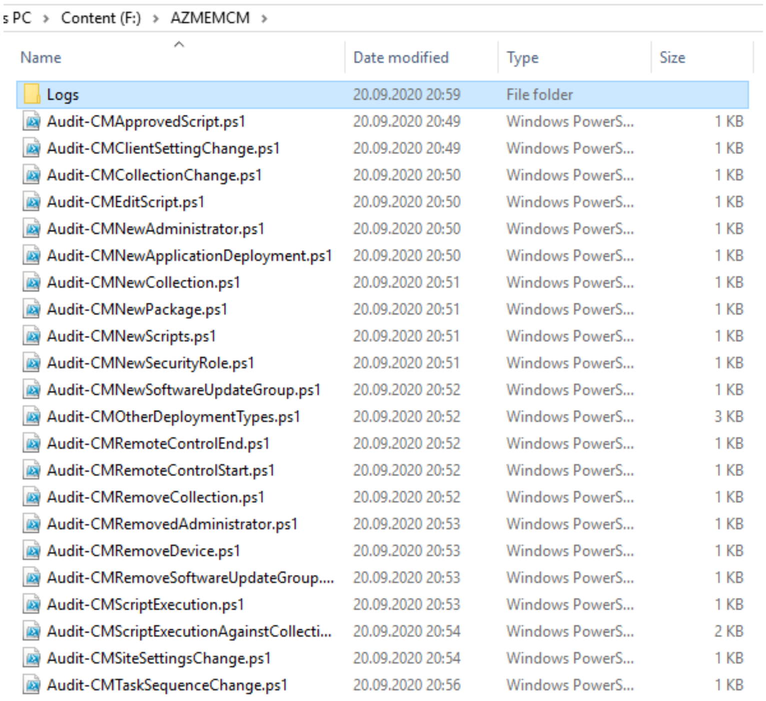
PSEventViewer PowerShell module installation

Still, in PowerShell ISE run the following:

```
Install-Module PSEventViewer -Force -Verbose
```


Download scripts from GitHub

Download the scripts from https://github.com/Kaidja/AZMEMCM address and save the Scripts under **F:\AZMEMCM** folder. For example



s PC > Content (F:) > AZMEMCM >

Name	Date modified	Type	Size
Logs	20.09.2020 20:59	File folder	
Audit-CMApprovedScript.ps1	20.09.2020 20:49	Windows PowerS...	1 KB
Audit-CMClientSettingChange.ps1	20.09.2020 20:49	Windows PowerS...	1 KB
Audit-CMCollectionChange.ps1	20.09.2020 20:50	Windows PowerS...	1 KB
Audit-CMEditScript.ps1	20.09.2020 20:50	Windows PowerS...	1 KB
Audit-CMNewAdministrator.ps1	20.09.2020 20:50	Windows PowerS...	1 KB
Audit-CMNewApplicationDeployment.ps1	20.09.2020 20:50	Windows PowerS...	1 KB
Audit-CMNewCollection.ps1	20.09.2020 20:51	Windows PowerS...	1 KB
Audit-CMNewPackage.ps1	20.09.2020 20:51	Windows PowerS...	1 KB
Audit-CMNewScripts.ps1	20.09.2020 20:51	Windows PowerS...	1 KB
Audit-CMNewSecurityRole.ps1	20.09.2020 20:51	Windows PowerS...	1 KB
Audit-CMNewSoftwareUpdateGroup.ps1	20.09.2020 20:52	Windows PowerS...	1 KB
Audit-CMOtherDeploymentTypes.ps1	20.09.2020 20:52	Windows PowerS...	3 KB
Audit-CMRemoteControlEnd.ps1	20.09.2020 20:52	Windows PowerS...	1 KB
Audit-CMRemoteControlStart.ps1	20.09.2020 20:52	Windows PowerS...	1 KB
Audit-CMRemoveCollection.ps1	20.09.2020 20:52	Windows PowerS...	1 KB
Audit-CMRemovedAdministrator.ps1	20.09.2020 20:53	Windows PowerS...	1 KB
Audit-CMRemoveDevice.ps1	20.09.2020 20:53	Windows PowerS...	1 KB
Audit-CMRemoveSoftwareUpdateGroup....	20.09.2020 20:53	Windows PowerS...	1 KB
Audit-CMScriptExecution.ps1	20.09.2020 20:53	Windows PowerS...	1 KB
Audit-CMScriptExecutionAgainstCollecti...	20.09.2020 20:54	Windows PowerS...	2 KB
Audit-CMSiteSettingsChange.ps1	20.09.2020 20:54	Windows PowerS...	1 KB
Audit-CMTaskSequenceChange.ps1	20.09.2020 20:56	Windows PowerS...	1 KB

You should have 22 scripts under **F:\AZMEMCM** folder.

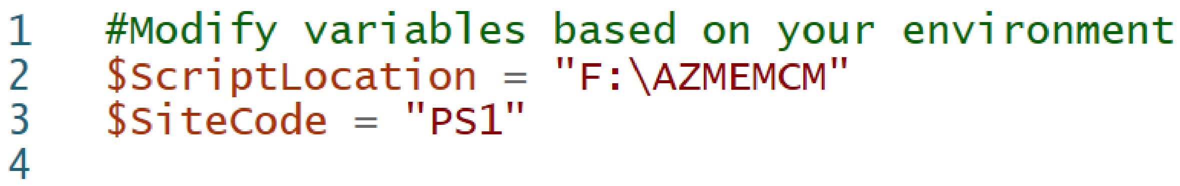
	Create **Logs** folder the main root folder

Create Status Filter Rules

	On your Primary Site server, run **PowerShell ISE** as an administrator

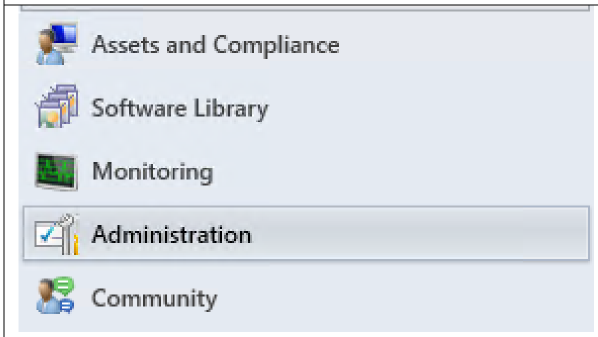
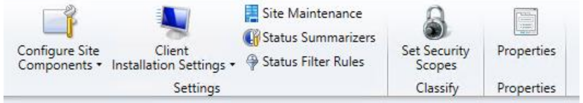
	Download **Create-CustoMStatusFilterRules.ps1** PowerShell script from GitHub - https://github.com/Kaidja/AZMEMCM
	Open **Create-CustoMStatusFilterRules.ps1** with PowerShell ISE

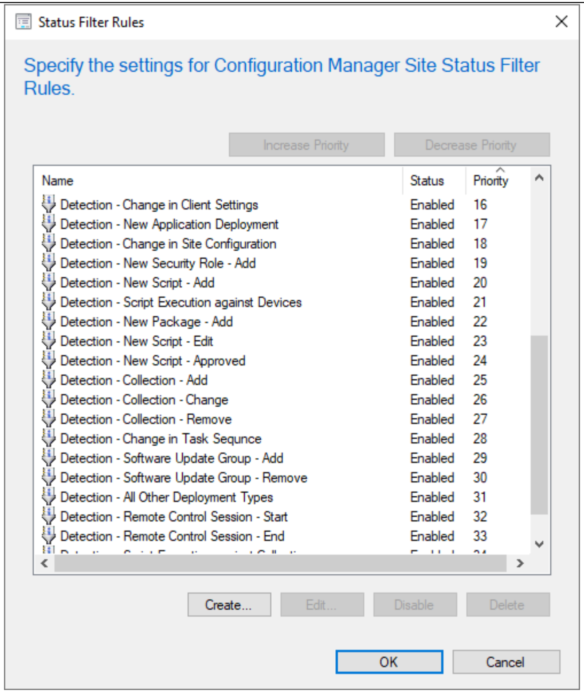
Modify the first two lines



```
#Modify variables based on your environment
$ScriptLocation = "F:\AZMEMCM"
$SiteCode = "PS1"

```

	Run the script using F5 or pressing the Run Script button
 Assets and Compliance Software Library Monitoring Administration Community	Open Configuration Manager Admin console and browse to the following workspace: - **Administration** - > **Site Configuration** -> **Sites**
 Configure Site Components Client Installation Settings Site Maintenance Status Summarizers Status Filter Rules Settings Set Security Scopes Classify Properties Properties	Select **Status Filter Rules**



Status Filter Rules

Specify the settings for Configuration Manager Site Status Filter Rules.

Name	Status	Priority
Detection - Change in Client Settings	Enabled	16
Detection - New Application Deployment	Enabled	17
Detection - Change in Site Configuration	Enabled	18
Detection - New Security Role - Add	Enabled	19
Detection - New Script - Add	Enabled	20
Detection - Script Execution against Devices	Enabled	21
Detection - New Package - Add	Enabled	22
Detection - New Script - Edit	Enabled	23
Detection - New Script - Approved	Enabled	24
Detection - Collection - Add	Enabled	25
Detection - Collection - Change	Enabled	26
Detection - Collection - Remove	Enabled	27
Detection - Change in Task Sequnce	Enabled	28
Detection - Software Update Group - Add	Enabled	29
Detection - Software Update Group - Remove	Enabled	30
Detection - All Other Deployment Types	Enabled	31
Detection - Remote Control Session - Start	Enabled	32
Detection - Remote Control Session - End	Enabled	33

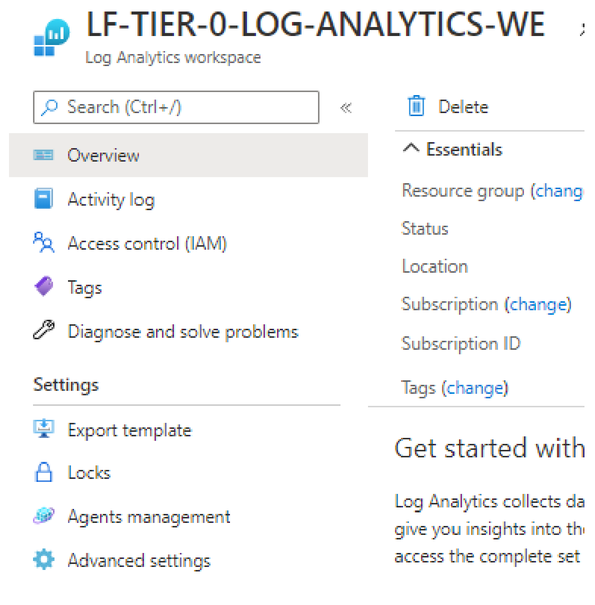
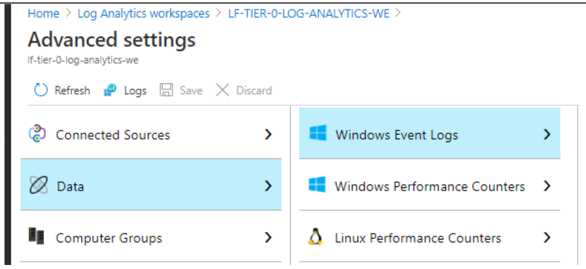
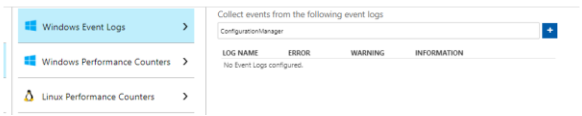
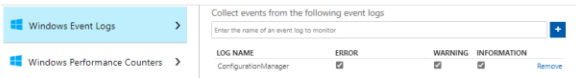
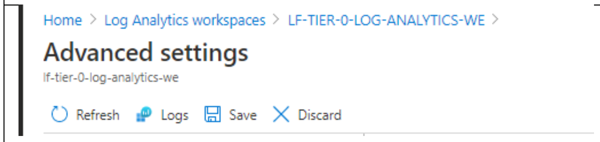
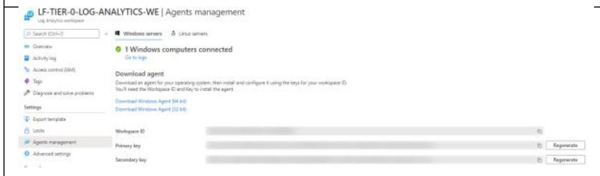
You should have **22** different rules.

Create Log Analytics workspace

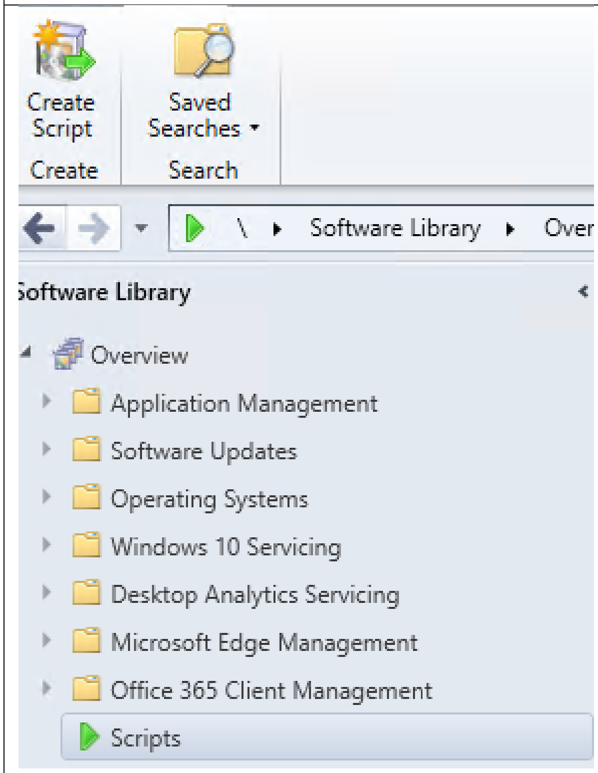
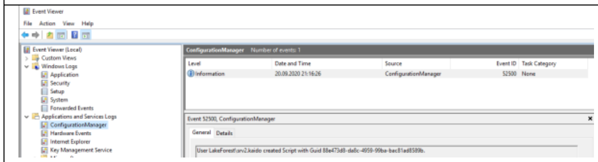
Follow the following steps to create the Log Analytics workspace

Configure Log Analytics workspace

Open **portal.azure.com** and select your newly created Log Analytics workspace

 LF-TIER-0-LOG-ANALYTICS-WE Log Analytics workspace Search (Ctrl+/) Delete Overview Activity log Access control (IAM) Tags Diagnose and solve problems Settings Export template Locks Agents management Advanced settings Essentials Resource group (chang Status Location Subscription (change) Subscription ID Tags (change) Get started with Log Analytics collects da give you insights into th access the complete set	Select **Advanced Settings**
 Home > Log Analytics workspaces > LF-TIER-0-LOG-ANALYTICS-WE > Advanced settings lf-tier-0-log-analytics-we Refresh Logs Save Discard Connected Sources Data Computer Groups Windows Event Logs Windows Performance Counters Linux Performance Counters	Select **Data** -> **Windows Event Logs**
 Windows Event Logs Windows Performance Counters Linux Performance Counters Collect events from the following event logs ConfigurationManager LOG NAME ERROR WARNING INFORMATION No Event Logs configured.	Enter **ConfigurationManager** as a log name and click the **+** sign
 Collect events from the following event logs Enter the name of an event log to monitor LOG NAME ERROR WARNING INFORMATION ConfigurationManager ☑ ☑ ☑ Remove	You should see the **ConfigurationManager** log name
 Home > Log Analytics workspaces > LF-TIER-0-LOG-ANALYTICS-WE > Advanced settings lf-tier-0-log-analytics-we Refresh Logs Save Discard	Click **Save**
 LF-TIER-0-LOG-ANALYTICS-WE | Agents management Log Analytics workspace Search (Ctrl+/) Overview Activity log Access control (IAM) Tags Diagnose and solve problems Settings Export template Locks Agents management Advanced settings Windows servers Linux servers Go to logs Download Windows Agent (64 bit) Download Windows Agent (32 bit) Regenerate Regenerate 1 Windows computers connected Download agent Download an agent for your operating system, then install and configure it using the keys for your workspace ID. You'll need the Workspace ID and Key to install the agent. Workspace ID Primary key Secondary key	Go back to the main dashboard and select **Agent Management.**
	Download the 64-bit agent and install the agent on your Primary Site Server. During the installation, enter the **Workspace ID** and **Primary Key**.

Testing

	Open Configuration Manager admin console and browse to the following workspace **Software Library** -> **Scripts**
 Create Script Create Saved Searches Search \ ▸ Software Library ▸ Over Software Library Overview Application Management Software Updates Operating Systems Windows 10 Servicing Desktop Analytics Servicing Microsoft Edge Management Office 365 Client Management Scripts	Create a new script called **Collect-TestLogs** and add some random text. Finish the new script wizard with default settings.
 Event Viewer File Action View Help Event Viewer (Local) Custom Views Windows Logs Application Security Setup System Forwarded Events Applications and Services Logs ConfigurationManager Hardware Events Internet Explorer Key Management Service General Details ConfigurationManager Number of events: 1 Level: Information · Date and Time: 20.09.2020 21:16:26 · Source: ConfigurationManager · Event ID: 52500 · Task Category: None Event 52500, ConfigurationManager User LakeForest\srv2.kaido created Script with Guid 88e473d8-da8c-4959-99ba-bac81ad8589b.	After that, you should see the Event ID **52500**

Use cases and samples

Now that our environment is up and running, we can start playing around with Log Analytics queries.

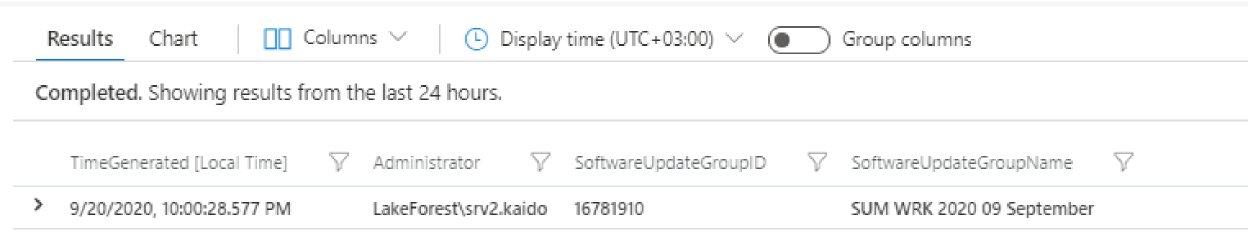
Software Updates

New Software Update Group

- Example KQL query

```
//New Software Update Group
Event
| where EventID == 30219 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend SoftwareUpdateGroupID = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| extend SoftwareUpdateGroupName = extractjson("$['DataItem']['EventData']['Data'][3]", MyData)
| project Administrator, SoftwareUpdateGroupID, SoftwareUpdateGroupName, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

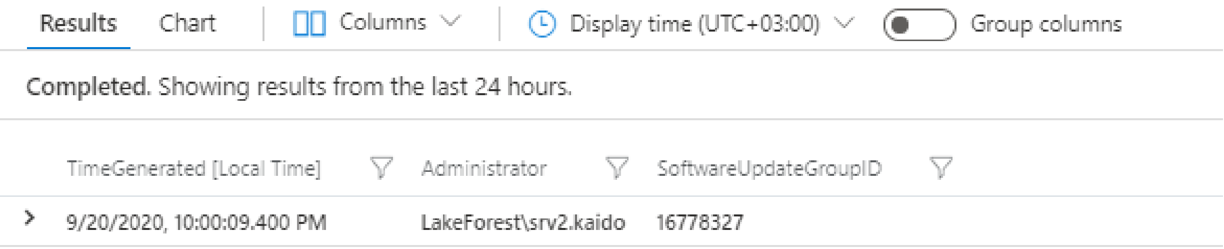
TimeGenerated [Local Time]	Administrator	SoftwareUpdateGroupID	SoftwareUpdateGroupName
9/20/2020, 10:00:28.577 PM	LakeForest\srv2.kaido	16781910	SUM WRK 2020 09 September

Someone removed the Software Update Group

- Example KQL query

```
//Removal of Software Update Group
Event
| where EventID == 30221 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend SoftwareUpdateGroupID = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| project Administrator, SoftwareUpdateGroupID, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

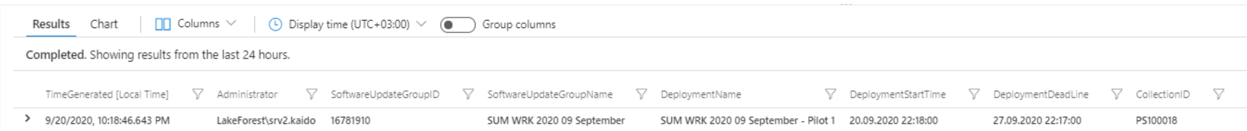
	TimeGenerated [Local Time]	Administrator	SoftwareUpdateGroupID
>	9/20/2020, 10:00:09.400 PM	LakeForest\srv2.kaido	16778327

Software Updates deployment

- Example KQL query

```
//Software Updates Deployments
Event
| where EventID == 30196 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend SoftwareUpdateGroupID = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| extend SoftwareUpdateGroupName = extractjson("$['DataItem']['EventData']['Data'][3]", MyData)
| extend DeploymentName = extractjson("$['DataItem']['EventData']['Data'][4]", MyData)
| extend DeploymentStartTime = extractjson("$['DataItem']['EventData']['Data'][5]", MyData)
| extend DeploymentDeadLine = extractjson("$['DataItem']['EventData']['Data'][6]", MyData)
| extend CollectionID = extractjson("$['DataItem']['EventData']['Data'][7]", MyData)
| extend Featuretype = extractjson("$['DataItem']['EventData']['Data'][8]", MyData)
| where Featuretype == "Software Update"
| project Administrator, SoftwareUpdateGroupID,
SoftwareUpdateGroupName,DeploymentName,DeploymentStartTime,DeploymentDeadLine,Collectio
nID, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

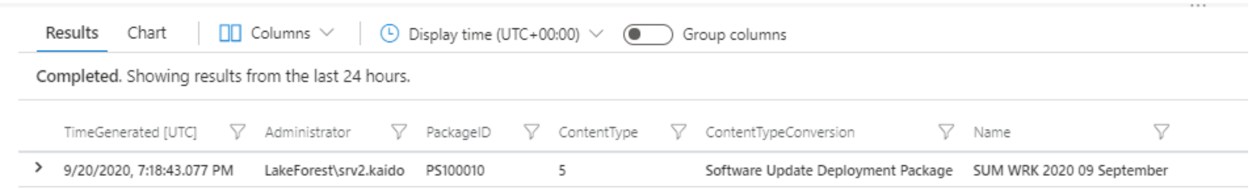
	TimeGenerated [Local Time]	Administrator	SoftwareUpdateGroupID	SoftwareUpdateGroupName	DeploymentName	DeploymentStartTime	DeploymentDeadLine	CollectionID
>	9/20/2020, 10:18:46.643 PM	LakeForest\srv2.kaido	16781910	SUM WRK 2020 09 September	SUM WRK 2020 09 September - Pilot 1	20.09.2020 22:18:00	27.09.2020 22:17:00	PS100018

Packages

- Example KQL query

```
//New package creation - generic ID for different package types
// Package IDs
// 0 - classic software package, 5 - Software Update Deployment Package, 4 - Task Sequence
Event
| where EventID == 30000 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend PackageID = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| extend ContentType = extractjson("$['DataItem']['EventData']['Data'][3]", MyData)
| extend ContentTypeConversion = case(ContentType == 0, "Package",ContentType == 5,
"Software Update Deployment Package",ContentType == 4, "Task Sequence","unknown")
| extend Name = extractjson("$['DataItem']['EventData']['Data'][4]", MyData)
| project Administrator, PackageID, ContentType, ContentTypeConversion, Name, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

TimeGenerated [UTC]	Administrator	PackageID	ContentType	ContentTypeConversion	Name
9/20/2020, 7:18:43.077 PM	LakeForest\srv2.kaido	PS100010	5	Software Update Deployment Package	SUM WRK 2020 09 September

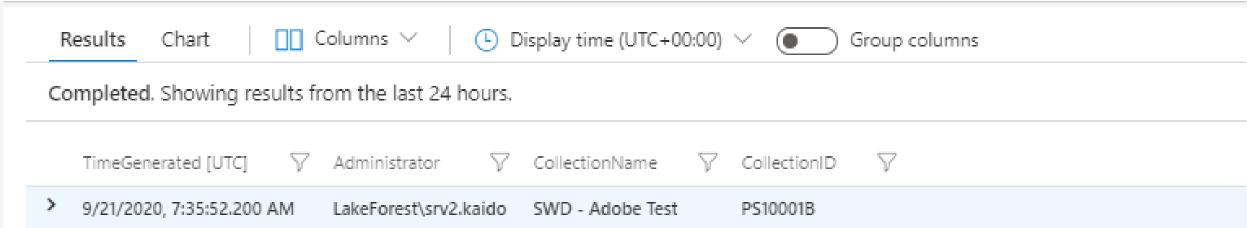
Collections

Collection creation

- Example KQL query

```
//New Collection in Configuration Manager
Event
| where EventID == 30015 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend CollectionName = extractjson("$['DataItem']['EventData']['Data'][3]", MyData)
| extend CollectionID = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| project Administrator, CollectionName, CollectionID, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

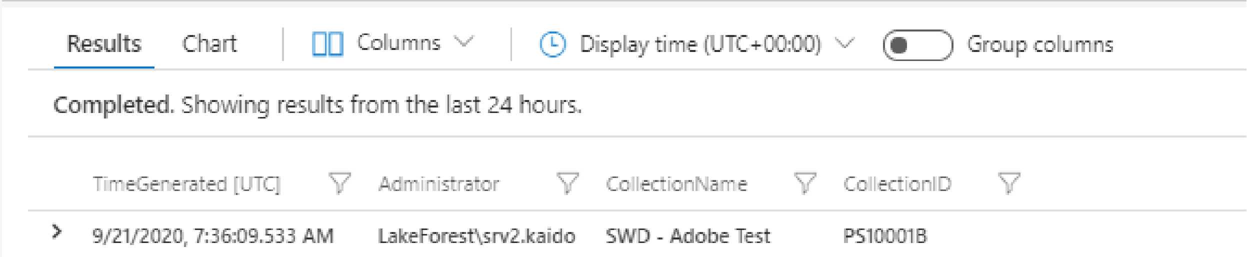
	TimeGenerated [UTC]	Administrator	CollectionName	CollectionID
>	9/21/2020, 7:35:52.200 AM	LakeForest\srv2.kaido	SWD - Adobe Test	PS10001B

Collection removal

- Example KQL query

```
//Collection removed
Event
| where EventID == 30017 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend CollectionName = extractjson("$['DataItem']['EventData']['Data'][3]", MyData)
| extend CollectionID = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| project Administrator, CollectionName, CollectionID, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

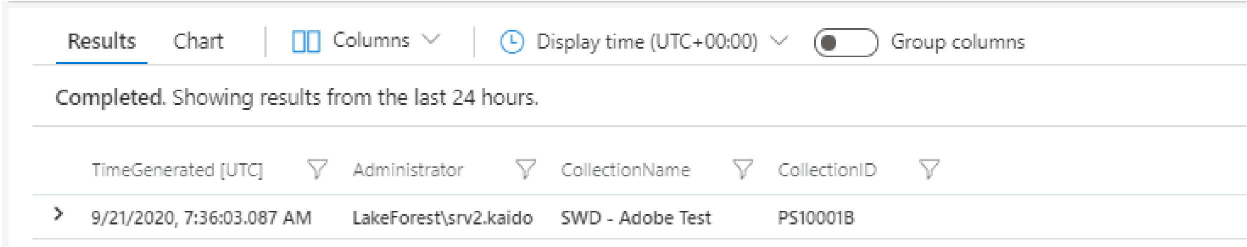
	TimeGenerated [UTC]	Administrator	CollectionName	CollectionID
>	9/21/2020, 7:36:09.533 AM	LakeForest\srv2.kaido	SWD - Adobe Test	PS10001B

Collection change

- Example KQL query

```
//Collection Change in Configuration Manager
Event
| where EventID == 30016 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend CollectionName = extractjson("$['DataItem']['EventData']['Data'][3]", MyData)
| extend CollectionID = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| project Administrator, CollectionName, CollectionID, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

TimeGenerated [UTC]	Administrator	CollectionName	CollectionID
9/21/2020, 7:36:03.087 AM	LakeForest\srv2.kaido	SWD - Adobe Test	PS10001B

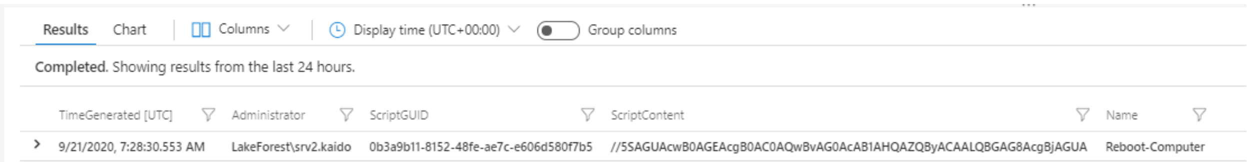
Scripts

New Script

- Example KQL query

```
//New Script in Configuration Manager
Event
| where EventID == 52500 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend ScriptGUID = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| extend ScriptContent = extractjson("$['DataItem']['EventData']['Data'][4]", MyData)
| extend Name = extractjson("$['DataItem']['EventData']['Data'][5]", MyData)
| project Administrator, ScriptGUID, ScriptContent, Name, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

TimeGenerated [UTC]	Administrator	ScriptGUID	ScriptContent	Name
9/21/2020, 7:28:30.553 AM	LakeForest\srv2.kaido	0b3a9b11-8152-48fe-ae7c-e606d580f7b5	//5SAGUAcwB0AGEAcgB0AC0AQwBvAG0AcAB1AHQAZQByACAALQBGAG8AcgBjAGUA	Reboot-Computer

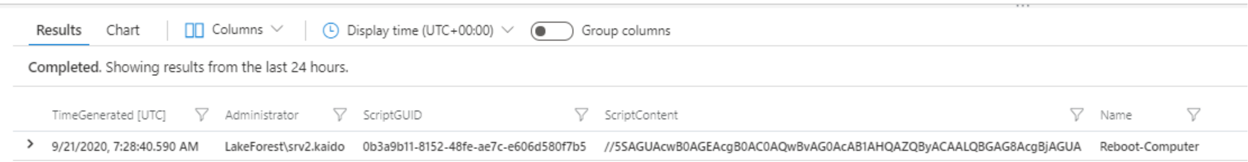
ScirptContent holds the actual script from the Configuration Manager. You can decode the script.

Script approved

- Example KQL query

```
//Script approved in Configuration Manager
Event
| where EventID == 52501 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend ScriptGUID = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| extend ScriptContent = extractjson("$['DataItem']['EventData']['Data'][4]", MyData)
| extend Name = extractjson("$['DataItem']['EventData']['Data'][5]", MyData)
| project Administrator, ScriptGUID, ScriptContent, Name, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

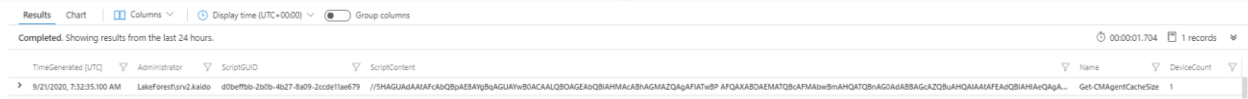
	TimeGenerated [UTC]	Administrator	ScriptGUID	ScriptContent	Name
›	9/21/2020, 7:28:40.590 AM	LakeForest\srv2.kaido	0b3a9b11-8152-48fe-ae7c-e606d580f7b5	//5SAGUAcwB0AGEAcgB0AC0AQwBvAG0AcAB1AHQAZQByACAALQBGAG8AcgBjAGUA	Reboot-Computer

Script execution against devices

- Example KQL query

```
//Script execution against devices
Event
| where EventID == 40806 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend ScriptGUID = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| extend ScriptContent = extractjson("$['DataItem']['EventData']['Data'][4]", MyData)
| extend Name = extractjson("$['DataItem']['EventData']['Data'][5]", MyData)
| extend DeviceCount = extractjson("$['DataItem']['EventData']['Data'][6]", MyData)
| project Administrator, ScriptGUID, ScriptContent, Name, DeviceCount, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

00:00:01.704 1 records

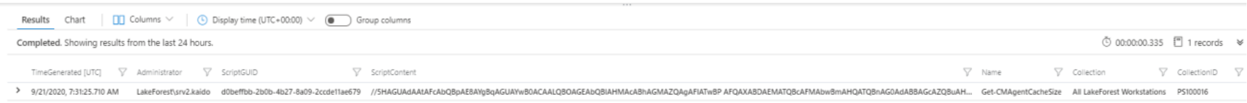
	TimeGenerated [UTC]	Administrator	ScriptGUID	ScriptContent	Name	DeviceCount
›	9/21/2020, 7:32:35.100 AM	LakeForest\srv2.kaido	d0beffbb-2b0b-4b27-8a09-2ccde11ae679	//5HAGUAdAAtAFcAbQBpAE8AYgBqAGUAYwB0ACAALQBOAGEAbQBlAHMAcABhAGMAZQAgAFIATwBP AFQAXABDAEMATQBcAFMAbwBmAHQATQBnAG0AdABBAGcAZQBuAHQAIAAtAFEAdQBlAHIAeQAgA...	Get-CMAgentCacheSize	1

Script execution against Collection

- Example KQL query

```
//Script execution against Collection
Event
| where EventID == 40805 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend ScriptGUID = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| extend ScriptContent = extractjson("$['DataItem']['EventData']['Data'][4]", MyData)
| extend Name = extractjson("$['DataItem']['EventData']['Data'][5]", MyData)
| extend Collection = extractjson("$['DataItem']['EventData']['Data'][6]", MyData)
| extend CollectionID = extractjson("$['DataItem']['EventData']['Data'][7]", MyData)
| project Administrator, ScriptGUID, ScriptContent, Name, Collection, CollectionID, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

00:00:00.335 1 records

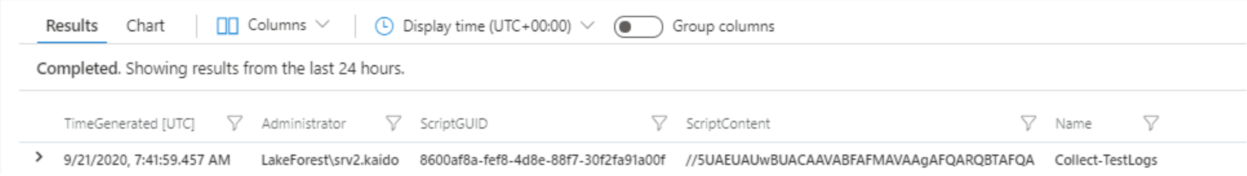
TimeGenerated [UTC]	Administrator	ScriptGUID	ScriptContent	Name	Collection	CollectionID
9/21/2020, 7:31:25.710 AM	LakeForest\srv2.kaido	d0beffbb-2b0b-4b27-8a09-2ccde11ae679	//5HAGUAdAAtAFcAbQBpAE8AYgBqAGUAYwB0ACAALQBOAGEAbQBIAHMAcABhAGMAZQAgAFIATwBP AFQAXABDAEMATQBcAFMAbwBmAHQATQBnAG0AdABBAGcAZQBuAH...	Get-CMAgentCacheSize	All LakeForest Workstations	PS100016

Script change

- Example KQL query

```
//Edit Script in Configuration Manager
Event
| where EventID == 52506 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend ScriptGUID = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| extend ScriptContent = extractjson("$['DataItem']['EventData']['Data'][4]", MyData)
| extend Name = extractjson("$['DataItem']['EventData']['Data'][5]", MyData)
| project Administrator, ScriptGUID, ScriptContent, Name, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

TimeGenerated [UTC]	Administrator	ScriptGUID	ScriptContent	Name
9/21/2020, 7:41:59.457 AM	LakeForest\srv2.kaido	8600af8a-fef8-4d8e-88f7-30f2fa91a00f	//5UAEUAUwBUACAAVABFAFMAVAAgAFQARQBTAFQA	Collect-TestLogs

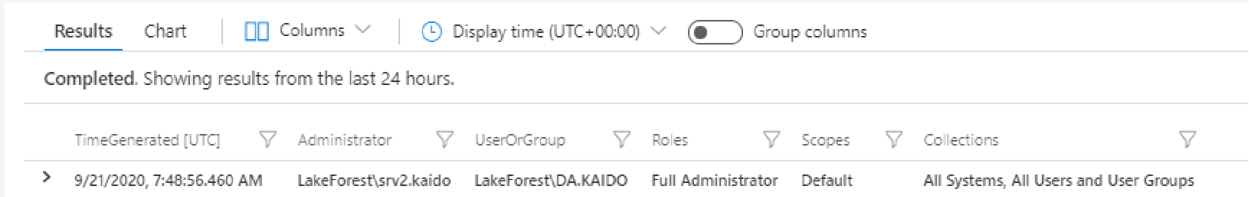
Security

New Administrator

- Example KQL query

```
//Somebody added a new user  / group - EventID 31240
Event
| where EventID == 31240 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend UserOrGroup = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| extend Roles = extractjson("$['DataItem']['EventData']['Data'][3]", MyData)
| extend Scopes = extractjson("$['DataItem']['EventData']['Data'][4]", MyData)
| extend Collections = extractjson("$['DataItem']['EventData']['Data'][5]", MyData)
| project Administrator, UserOrGroup, Roles, Scopes, Collections, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

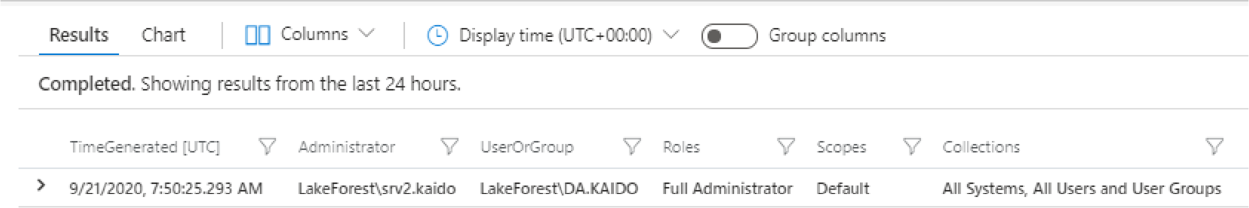
	TimeGenerated [UTC]	Administrator	UserOrGroup	Roles	Scopes	Collections
>	9/21/2020, 7:48:56.460 AM	LakeForest\srv2.kaido	LakeForest\DA.KAIDO	Full Administrator	Default	All Systems, All Users and User Groups

Administrator removed

- Example KQL query

```
//Somebody removed a user or security group
Event
| where EventID == 31242 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend UserOrGroup = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| extend Roles = extractjson("$['DataItem']['EventData']['Data'][3]", MyData)
| extend Scopes = extractjson("$['DataItem']['EventData']['Data'][4]", MyData)
| extend Collections = extractjson("$['DataItem']['EventData']['Data'][5]", MyData)
| project Administrator, UserOrGroup, Roles, Scopes, Collections, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

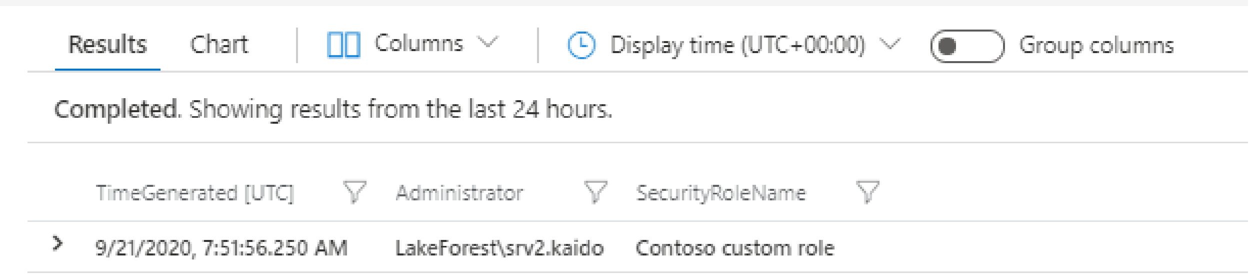
	TimeGenerated [UTC]	Administrator	UserOrGroup	Roles	Scopes	Collections
>	9/21/2020, 7:50:25.293 AM	LakeForest\srv2.kaido	LakeForest\DA.KAIDO	Full Administrator	Default	All Systems, All Users and User Groups

New security role

- Example KQL query

```
//New Security Role
Event
| where EventID == 31200 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend SecurityRoleName = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| project Administrator, SecurityRoleName, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

	TimeGenerated [UTC]	Administrator	SecurityRoleName
>	9/21/2020, 7:51:56.250 AM	LakeForest\srv2.kaido	Contoso custom role

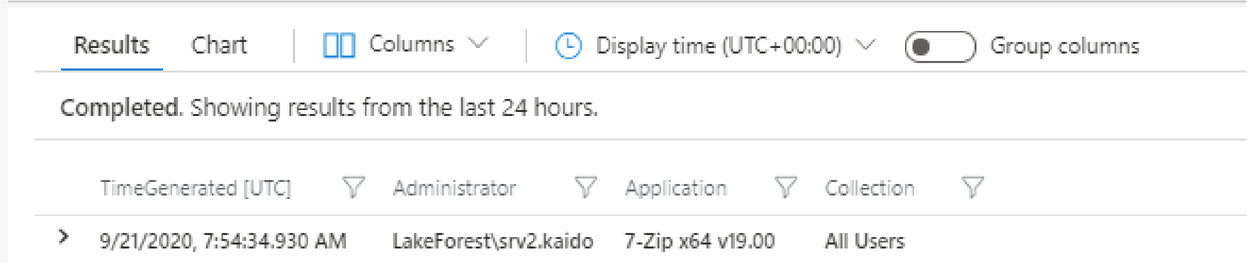
Applications

New Application deployment

- Example KQL query

```
//New Application deployment - EventID 30226
Event
| where EventID == 30226 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend Application = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| extend Collection = extractjson("$['DataItem']['EventData']['Data'][3]", MyData)
| project Administrator, Application, Collection, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

	TimeGenerated [UTC]	Administrator	Application	Collection
>	9/21/2020, 7:54:34.930 AM	LakeForest\srv2.kaido	7-Zip x64 v19.00	All Users

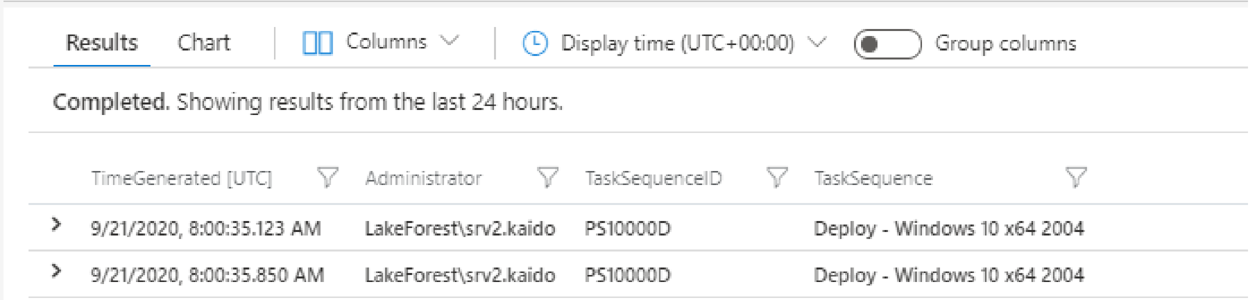
Operating system deployment

Change in Task Sequence

- Example KQL query

```
//Change in Task Sequence
Event
| where EventID == 30001 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend TaskSequenceID = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| extend TaskSequence = extractjson("$['DataItem']['EventData']['Data'][3]", MyData)
| project Administrator, TaskSequenceID, TaskSequence, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

	TimeGenerated [UTC]	Administrator	TaskSequenceID	TaskSequence
>	9/21/2020, 8:00:35.123 AM	LakeForest\srv2.kaido	PS10000D	Deploy - Windows 10 x64 2004
>	9/21/2020, 8:00:35.850 AM	LakeForest\srv2.kaido	PS10000D	Deploy - Windows 10 x64 2004

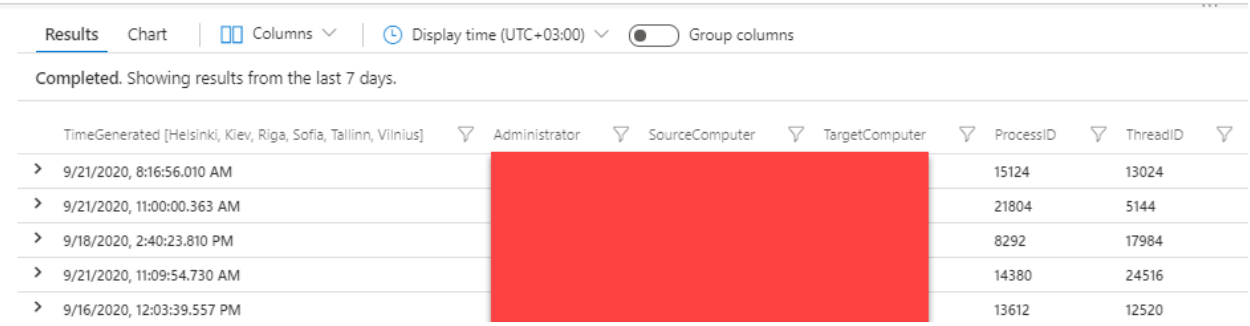
Remote Control

Session start

- Example KQL query

```
//Remote Control Start session
Event
| where EventID == 30076 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend SourceComputer = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| extend TargetComputer = extractjson("$['DataItem']['EventData']['Data'][3]", MyData)
| extend ProcessID = extractjson("$['DataItem']['EventData']['Data'][4]", MyData)
| extend ThreadID = extractjson("$['DataItem']['EventData']['Data'][5]", MyData)
| project Administrator, SourceComputer, TargetComputer, ProcessID, ThreadID, TimeGenerated
```

- Results



Completed. Showing results from the last 7 days.

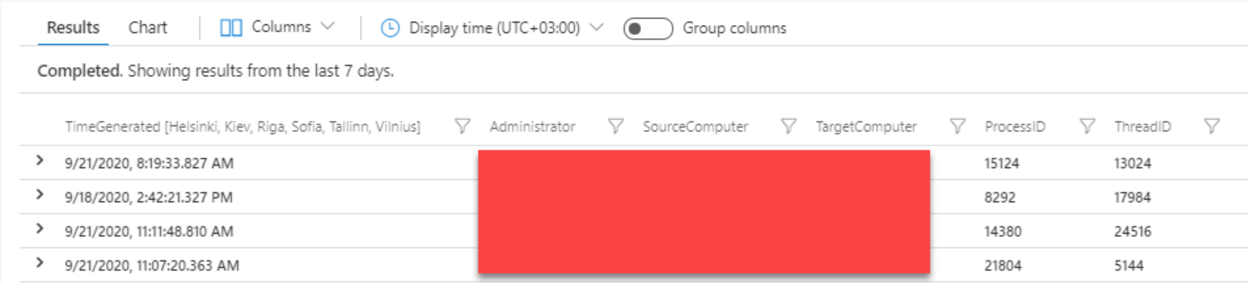
TimeGenerated [Helsinki, Kiev, Riga, Sofia, Tallinn, Vilnius]	Administrator	SourceComputer	TargetComputer	ProcessID	ThreadID
9/21/2020, 8:16:56.010 AM	[redacted]	[redacted]	[redacted]	15124	13024
9/21/2020, 11:00:00.363 AM	[redacted]	[redacted]	[redacted]	21804	5144
9/18/2020, 2:40:23.810 PM	[redacted]	[redacted]	[redacted]	8292	17984
9/21/2020, 11:09:54.730 AM	[redacted]	[redacted]	[redacted]	14380	24516
9/16/2020, 12:03:39.557 PM	[redacted]	[redacted]	[redacted]	13612	12520

Session end

- Example KQL query

```
//Remote Control end session
Event
| where EventID == 30077 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend SourceComputer = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| extend TargetComputer = extractjson("$['DataItem']['EventData']['Data'][3]", MyData)
| extend ProcessID = extractjson("$['DataItem']['EventData']['Data'][4]", MyData)
| extend ThreadID = extractjson("$['DataItem']['EventData']['Data'][5]", MyData)
| project Administrator, SourceComputer, TargetComputer, ProcessID, ThreadID, TimeGenerated
```

- Results



Completed. Showing results from the last 7 days.

TimeGenerated [Helsinki, Kiev, Riga, Sofia, Tallinn, Vilnius]	Administrator	SourceComputer	TargetComputer	ProcessID	ThreadID
9/21/2020, 8:19:33.827 AM	[redacted]	[redacted]	[redacted]	15124	13024
9/18/2020, 2:42:21.327 PM	[redacted]	[redacted]	[redacted]	8292	17984
9/21/2020, 11:11:48.810 AM	[redacted]	[redacted]	[redacted]	14380	24516
9/21/2020, 11:07:20.363 AM	[redacted]	[redacted]	[redacted]	21804	5144

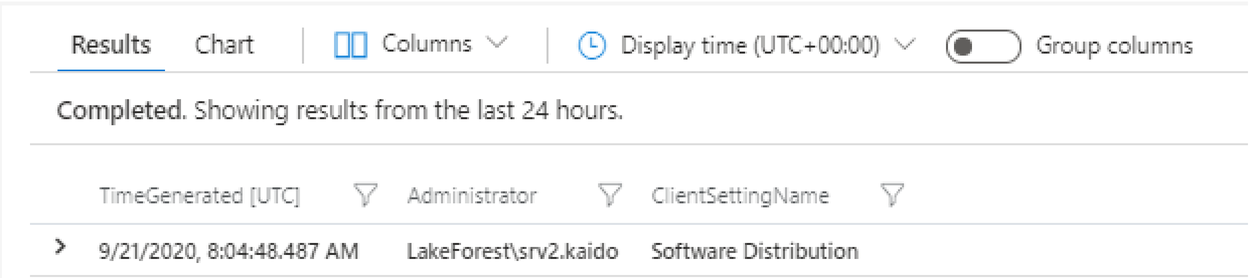
Site Configuration

Client Settings change

- Example KQL query

```
//Somebody changed the Client Settings
Event
| where EventID == 30043 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend ClientSettingName = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| project Administrator, ClientSettingName, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

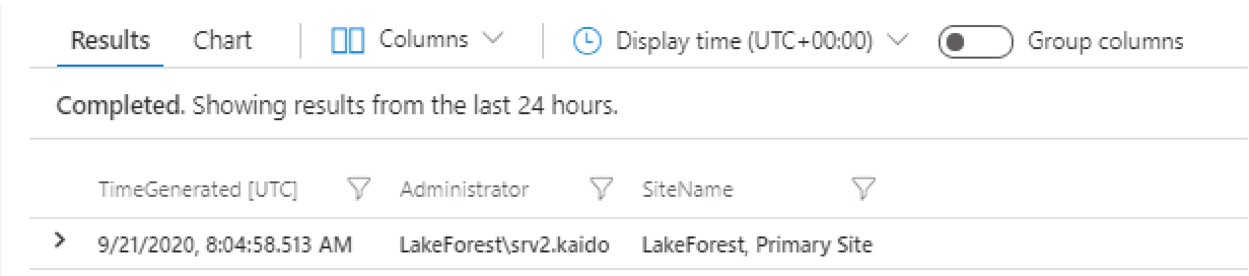
TimeGenerated [UTC]	Administrator	ClientSettingName
9/21/2020, 8:04:48.487 AM	LakeForest\srv2.kaido	Software Distribution

Hierarchy changes

- Example KQL query

```
//Change in Primary Site Settings
Event
| where EventID == 30031 and EventLog == "ConfigurationManager"
| extend MyData = tostring(parse_xml(EventData))
| extend Administrator = extractjson("$['DataItem']['EventData']['Data'][1]", MyData)
| extend SiteName = extractjson("$['DataItem']['EventData']['Data'][2]", MyData)
| project Administrator, SiteName, TimeGenerated
```

- Results



Completed. Showing results from the last 24 hours.

TimeGenerated [UTC]	Administrator	SiteName
9/21/2020, 8:04:58.513 AM	LakeForest\srv2.kaido	LakeForest, Primary Site