

PowerShell Sensor - Office365 / Exchange

Introduction

With the MS Powershell Sensor you are able to trigger any *Get-* cmdlet within your Office 365 tenant by default (Exchange and Security / Compliance)
By default we have restricted the cmdlets in a way, that only *GET-* and *TEST-* cmdlets are allowed. However you can adjust this by modifying the following file on the GL filesystem level.

for Exchange:

open ssh console and issue the following commands:

- `vim /opt/panagenda/appdata/volumes/gl/scripts/gl_powershell/o365_exchange_ps_commands.txt`
- Just add your cmdlet at the end of this list save/close.
- Afterwards please restart the following docker container: `docker restart gl_tomcat`
- Wait until the GFL application becomes available again (login screen)
- From now on you can choose your cmdlet entry within the sensor

for Security and Compliance

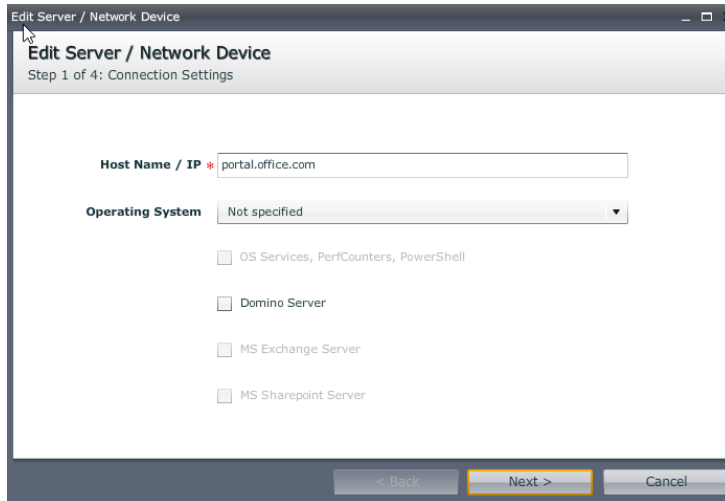
`vim /opt/panagenda/appdata/volumes/gl/scripts/gl_powershell/o365_sec_and_compliance_ps_commands.txt`

The following kbase article explains two simple examples for Exchange

1. Get-Mailbox
2. Get-MailTrafficReport

Configuration

First of all you have to add *portal.office.com* as a server to your GL server list. Just add the server without selecting any OS type or Role



Scenario 1: Get-Mailbox for a specific user

Create a *MS Powershell Sensor* with the following Settings

- Script Type: O365
- Command Type: Exchange
- Username+Password
- cmdlet: enter "Get-"....and choose Get-Mailbox from the List
- add your Parameters (make sure that you have one parameter PER line!)

Name: O365 - Get-Mailbox | Get-MailboxStatistics

Enabled [Show Schedule](#)

Settings | Targets | Actions | Schedule

Script Type: O365 | Username: *

Command Type: Exchange | Password: *

cmdlet Name: * Get-Mailbox

Parameters: -Identity | Get-MailboxStatistics

Only one parameter or parameter-value pair per line

Please make sure that you have assigned the correct Server Roles on the Node Level

Save & Close | Discard Changes

- add portal.office.com as a target (target tab)

Output:

greenlight.powershell.o365exchange.1.Identity.MapIdentity	
greenlight.powershell.o365exchange.1.Identity.ToString	a055789c-5a69-42fe-9c3e-e405df6550t
greenlight.powershell.o365exchange.1.ItemCount	31.420
greenlight.powershell.o365exchange.1.MailboxType.MailboxType	0
greenlight.powershell.o365exchange.1.MailboxType.ToString	Private
greenlight.powershell.o365exchange.1.MailboxTypeDetail.MailboxTypeDetail	1
greenlight.powershell.o365exchange.1.MailboxTypeDetail.ToString	UserMailbox
greenlight.powershell.o365exchange.1.MessageTableAvailableSize.IsUnlimited	0
greenlight.powershell.o365exchange.1.MessageTableAvailableSize.ToString	36.25 MB (38,010,880 bytes)
greenlight.powershell.o365exchange.1.MessageTableAvailableSize.Value.ToString	36.25 MB (38,010,880 bytes)
greenlight.powershell.o365exchange.1.MessageTableTotalSize.IsUnlimited	0
greenlight.powershell.o365exchange.1.MessageTableTotalSize.ToString	1002 MB (1,050,804,224 bytes)
greenlight.powershell.o365exchange.1.MessageTableTotalSize.Value.ToString	1002 MB (1,050,804,224 bytes)

Scenario 2: Get-MailTrafficReport for a single day

Create a MS Powershell Sensor with the following Settings

- Script Type: O365
- Command Type: Exchange
- Username+Password
- cmdlet: enter "Get-".and choose Get-MailTrafficReport from the List
- add your Parameters (make sure that you have one parameter PER line!)

Name: O365 - Get-MailTrafficReport

Enabled [Show Schedule](#)

Settings | Targets | Actions | Schedule

Script Type: O365 | Username: [redacted]

Command Type: Exchange | Password: [redacted]

cmdlet Name: Get-MailTrafficReport

Parameters:


```
-Direction Inbound
-StartDate 01/31/2017
-EndDate 02/01/2017
```

Only one parameter or parameter-value pair per line

Please make sure that you have assigned the correct Server Roles on the Node Level

Save & Close | Discard Changes

Output:

greenlight.powershell.o365exchange.1.Direction	Inbound
greenlight.powershell.o365exchange.1.Domain	
greenlight.powershell.o365exchange.1.EndDate	-62.135.769.600.000
greenlight.powershell.o365exchange.1.EventSubType	
greenlight.powershell.o365exchange.1.EventType	BCL0
greenlight.powershell.o365exchange.1.Index	0
greenlight.powershell.o365exchange.1.MessageCount	45
greenlight.powershell.o365exchange.1.Organization	panagendaGmbH.onmicrosoft.com

Of course all this output can be further used for charting/alerting

example: inbound e-mails

