

Managing the Microsoft Security stack with PowerShell



Alex Verboon

Alex Verboon



<https://twitter.com/alexverboon>



<https://www.linkedin.com/in/verboonalex/>

GitHub

<https://github.com/alexverboon>



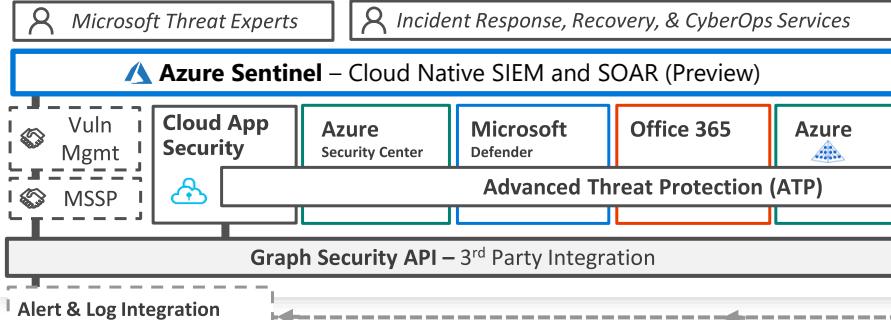
<https://www.verboon.info/>



Session Objectives

- Provide an overview of the various Microsoft Security Solutions and how you can interact with them using PowerShell
 - Windows Defender
 - Microsoft Defender Advanced Threat Protection
 - Microsoft Defender Exploit Guard
 - Microsoft Cloud App Security
 - Microsoft Office 365 Advanced Threat Protection
 - AzureAD Multifactor Authentication
- References and examples of useful PowerShell scripts / modules

Security Operations Center (SOC)



Cybersecurity Reference Architecture

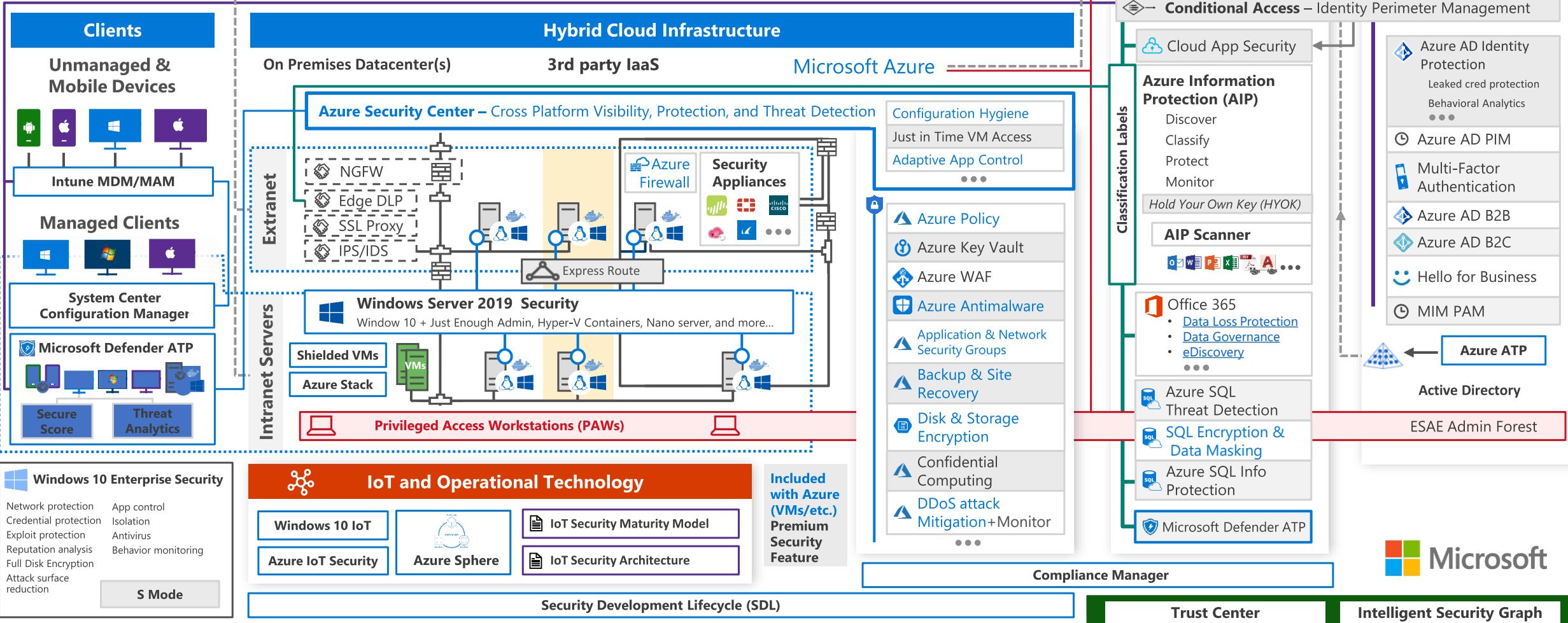
April 2019 – <https://aka.ms/MCRA> | Video Recording | Strategies

This is interactive!

1. Present Slide
2. Hover for Description
3. Click for more information

Roadmaps and Guidance

1. [Securing Privileged Access](#)
2. [Office 365 Security](#)
3. [Rapid Cyberattacks \(Wannacrypt/Petya\)](#)

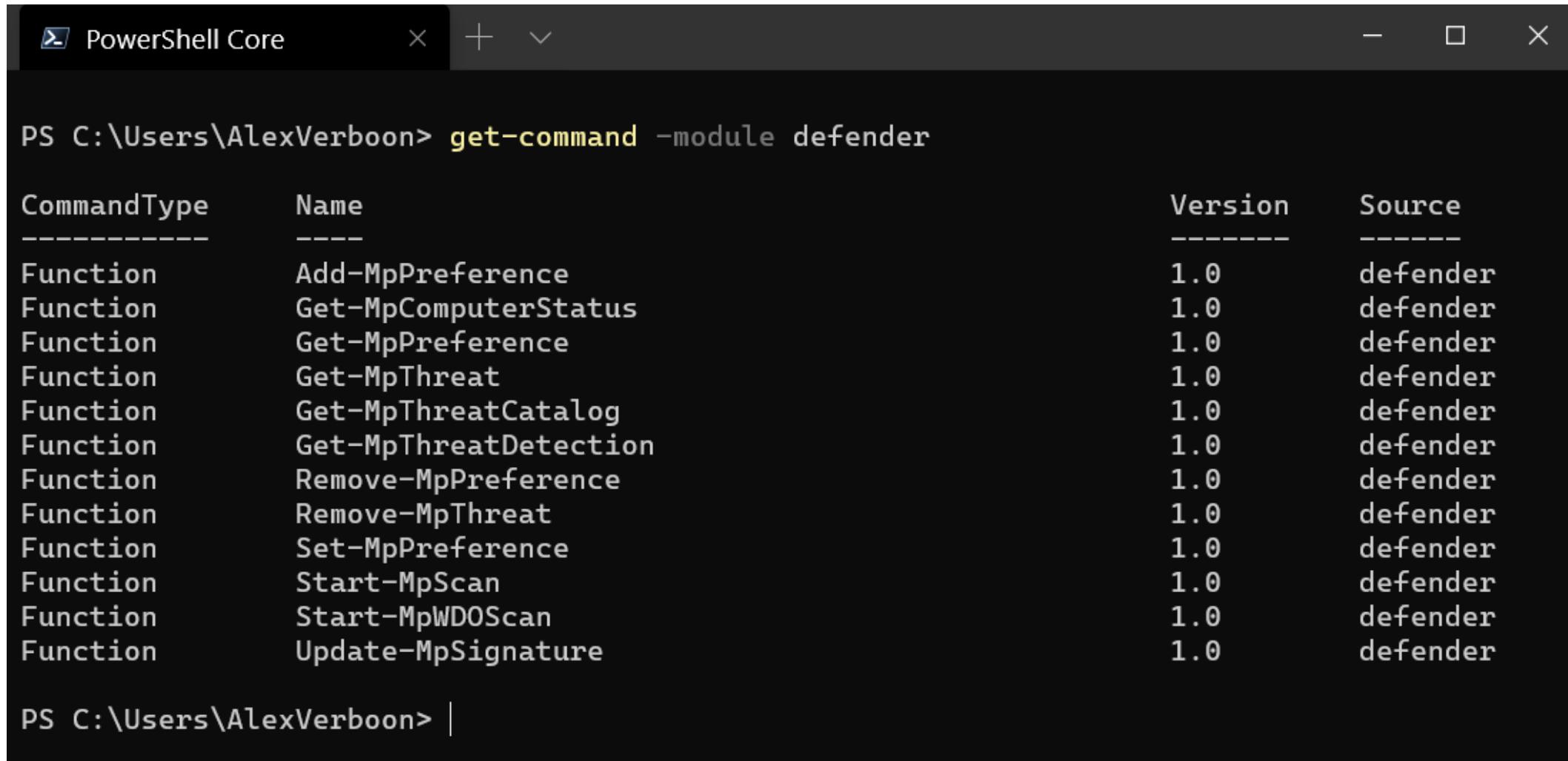




Windows Defender

Managing the Microsoft Security stack with PowerShell – Windows Defender

Windows Defender is installed by default on Windows 10 / Server 2016/2019

A screenshot of a PowerShell Core window titled "PowerShell Core". The window shows the output of the command "get-command -module defender". The output is a table with columns: CommandType, Name, Version, and Source. All commands listed are from the "defender" module and have a version of 1.0.

CommandType	Name	Version	Source
Function	Add-MpPreference	1.0	defender
Function	Get-MpComputerStatus	1.0	defender
Function	Get-MpPreference	1.0	defender
Function	Get-MpThreat	1.0	defender
Function	Get-MpThreatCatalog	1.0	defender
Function	Get-MpThreatDetection	1.0	defender
Function	Remove-MpPreference	1.0	defender
Function	Remove-MpThreat	1.0	defender
Function	Set-MpPreference	1.0	defender
Function	Start-MpScan	1.0	defender
Function	Start-MpWDOScan	1.0	defender
Function	Update-MpSignature	1.0	defender

```
PS C:\Users\AlexVerboon> get-command -module defender

 CommandType      Name          Version   Source
 -----          ----          -----   -----
 Function        Add-MpPreference    1.0      defender
 Function        Get-MpComputerStatus 1.0      defender
 Function        Get-MpPreference    1.0      defender
 Function        Get-MpThreat      1.0      defender
 Function        Get-MpThreatCatalog 1.0      defender
 Function        Get-MpThreatDetection 1.0      defender
 Function        Remove-MpPreference 1.0      defender
 Function        Remove-MpThreat    1.0      defender
 Function        Set-MpPreference    1.0      defender
 Function        Start-MpScan      1.0      defender
 Function        Start-MpWDOScan    1.0      defender
 Function        Update-MpSignature 1.0      defender

PS C:\Users\AlexVerboon> |
```

Managing the Microsoft Security stack with PowerShell – Windows Defender

```
PowerShell Core x + - □ ×  
PS C:\Users\AlexVerboon> Get-MpComputerStatus  
  
AMEngineVersion : 1.1.17300.4  
AMPProductVersion : 4.18.2006.10  
AMServiceEnabled : True  
AMServiceVersion : 4.18.2006.10  
AntispywareEnabled : True  
AntispywareSignatureAge : 0  
AntispywareSignatureLastUpdated : 8/1/2020 8:18:36 AM  
AntispywareSignatureVersion : 1.321.346.0  
AntivirusEnabled : True  
AntivirusSignatureAge : 0  
AntivirusSignatureLastUpdated : 8/1/2020 8:18:36 AM  
AntivirusSignatureVersion : 1.321.346.0  
BehaviorMonitorEnabled : True  
ComputerID : A5DB646A-B85D-4F7C-AB32-38B6CB060CB2  
ComputerState : 0  
FullScanAge : 178  
FullScanEndTime : 2/4/2020 3:26:33 PM  
FullScanStartTime : 2/4/2020 3:18:01 PM  
IoavProtectionEnabled : True  
IsTamperProtected : True  
IsVirtualMachine : False  
LastFullScanSource : 1  
LastQuickScanSource : 2  
NISEnabled : True  
NISEngineVersion : 1.1.17300.4  
NISSignatureAge : 0  
NISSignatureLastUpdated : 8/1/2020 8:18:36 AM  
NISSignatureVersion : 1.321.346.0  
OnAccessProtectionEnabled : True  
QuickScanAge : 9  
QuickScanEndTime : 7/23/2020 5:21:13 AM  
QuickScanStartTime : 7/23/2020 5:18:41 AM  
RealTimeProtectionEnabled : True  
RealTimeScanDirection : 0  
PSCo...  
PSComputerName :  
:
```

Use **Get-MpComputerStatus** to get an overview of installed versions and signature updates

```
PowerShell Core x + - □ ×  
PS C:\Users\AlexVerboon> $LastFullScanSource = @{  
    >>     Name = 'LastFullScanSource'  
    >>     Expression = {  
    >>         # property is an array, so process all values  
    >>         $value = $_.LastFullScanSource  
    >>  
    >>         switch(([int]$value)  
    >>             {  
    >>                 0          {'Unknown'}  
    >>                 1          {'User'}  
    >>                 2          {'System'}  
    >>                 3          {'Real-time'}  
    >>                 4          {'IOAV'}  
    >>             default      {"$value"}  
    >>         }  
    >>  
    >>     }  
    >> }  
PS C:\Users\AlexVerboon> Get-CimInstance -ClassName MSFT_MpComputerStatus -Namespace root/microsoft/windows/defender | Select-Object -Property Caption, $LastFullScanSource  
  
Caption LastFullScanSource  
-----  
User  
  
PS C:\Users\AlexVerboon> |
```

Managing the Microsoft Security stack with PowerShell – Windows Defender

Use **Get-MpPreference** to look at Windows Defender configuration settings

```
PS C:\Users\AlexVerboon> Get-MpPreference

AttackSurfaceReductionOnlyExclusions : {2, 1, 1, 1...}
AttackSurfaceReductionRules_Actions   : {01443614-cd74-433a-b99e-2ecdc07bfc25, 26190899-1602-49e8-8b27-eb1d0a1ce869, 3b576869-a4ec-4529-8536-b80a7769e899, 5beb7efe-fd9a-4556-801d-275e5ffc04cc...}
AttackSurfaceReductionRules_Ids      : False

CheckForSignaturesBeforeRunningScan : False
CloudBlockLevel                   : 2
CloudExtendedTimeout              : 50
ComputerID                        : A5DB646A-B85D-4F7C-AB32-38B6CB060CB2
ControlledFolderAccessAllowedApplications : {C:\Program Files\TechSmith\Snagit 2019\Snagit32.exe, C:\Program Files\TechSmith\Snagit 2019\SnagitEditor.exe, vmmms.exe, Vmsp.exe...}
```

Managing the Microsoft Security stack with PowerShell – Windows Defender

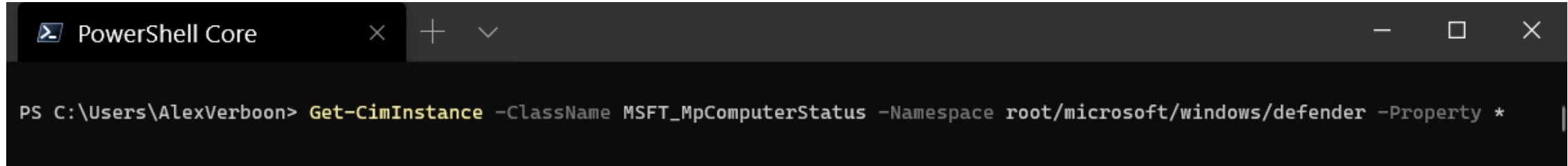
```
PowerShell Core -> + <- X

ExclusionExtension : {.avhd, .avhdx, .iso, .vhdx...}
ExclusionPath : {C:\AutomatedLab-VMs, C:\Dev\Public\SecLists, C:\LabSources\ISOs, C:\Program Files\WindowsPowerShell\Modules\AMSI\1.0.9...}
ExclusionProcess : {vmms.exe, Vmsp.exe, Vmwp.exe}
HighThreatDefaultAction : 0
LowThreatDefaultAction : 0
MAPSReporting : 2
MeteredConnectionUpdates : False
ModerateThreatDefaultAction : 0
PUAProtection : 1
QuarantinePurgeItemsAfterDelay : 90
RandomizeScheduleTaskTimes : True
RealTimeScanDirection : 0
RemediationScheduleDay : 0
RemediationScheduleTime : 02:00:00
ReportingAdditionalActionTimeOut : 10080
ReportingCriticalFailureTimeOut : 10080
ReportingNonCriticalTimeOut : 1440
ScanAvgCPULoadFactor : 50
ScanOnlyIfIdleEnabled : True
ScanParameters : 1
ScanPurgeItemsAfterDelay : 15
ScanScheduleDay : 0
ScanScheduleQuickScanTime : 00:00:00
ScanScheduleTime : 02:00:00
SevereThreatDefaultAction : 0
SharedSignaturesPath :
SignatureAuGracePeriod : 0
SignatureDefinitionUpdateFileSharesSources :
SignatureDisableUpdateOnStartupWithoutEngine : False
SignatureFallbackOrder : MicrosoftUpdateServer|MMPC
SignatureFirstAuGracePeriod : 120
SignatureScheduleDay : 8
SignatureScheduleTime : 01:45:00
SignatureUpdateCatchupInterval : 1
SignatureUpdateInterval : 4
SubmitSamplesConsent : 3
ThreatIDDefaultAction_Actions :
ThreatIDDefaultAction_Ids :
UILockdown : False
UnknownThreatDefaultAction : 0
PSComputerName :
```

PS C:\Users\AlexVerboon>

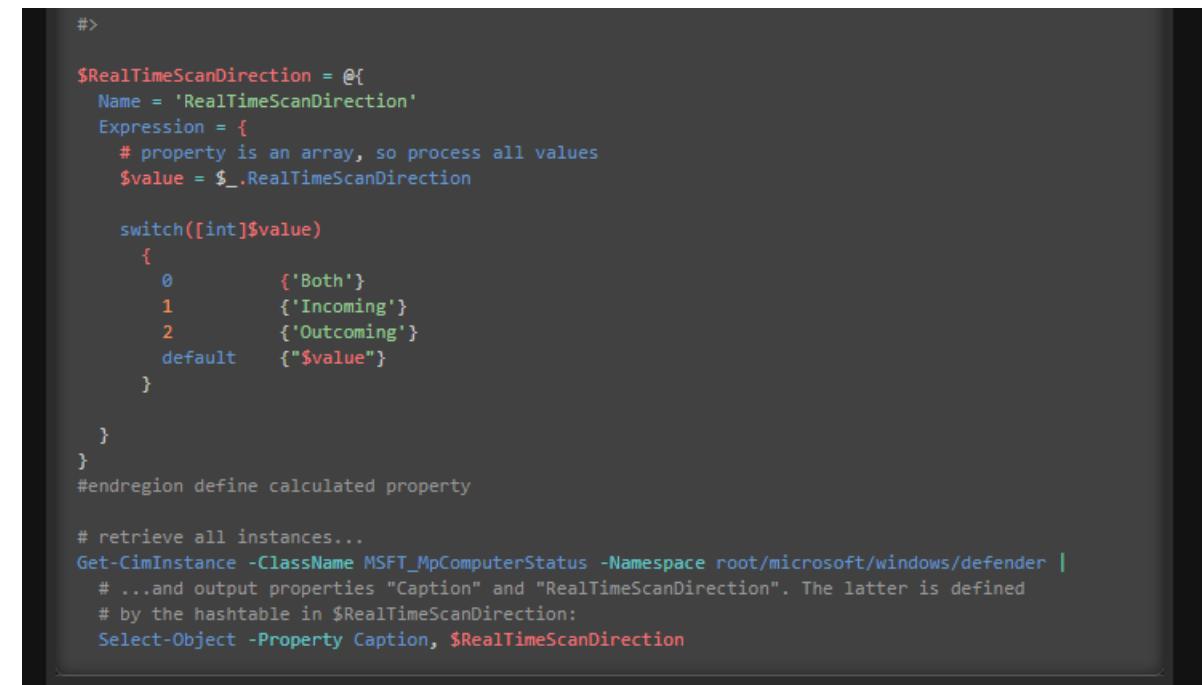
Managing the Microsoft Security stack with PowerShell – Windows Defender

Get-MpComputerStatus uses **MSFT_MpComputerStatus**



```
PS C:\Users\AlexVerboon> Get-CimInstance -ClassName MSFT_MpComputerStatus -Namespace root/microsoft/windows/defender -Property *
```

https://powershell.one/wmi/root/microsoft/windows/defender/msft_mpcomputerstatus



```
#>

$RealTimeScanDirection = @{
    Name = 'RealTimeScanDirection'
    Expression = {
        # property is an array, so process all values
        $value = $_.RealTimeScanDirection

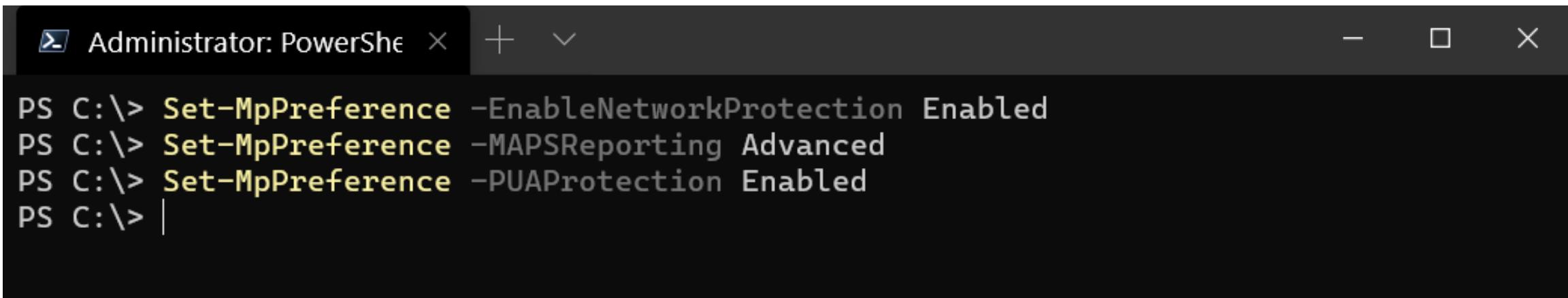
        switch(([int]$value)
        {
            0      {'Both'}
            1      {'Incoming'}
            2      {'Outcoming'}
            default {"$value"}
        })
    }
}

#endregion define calculated property

# retrieve all instances...
Get-CimInstance -ClassName MSFT_MpComputerStatus -Namespace root/microsoft/windows/defender |
    # ...and output properties "Caption" and "RealTimeScanDirection". The latter is defined
    # by the hashtable in $RealTimeScanDirection:
    Select-Object -Property Caption, $RealTimeScanDirection
```

Managing the Microsoft Security stack with PowerShell – Windows Defender

Use **Set-MpPreference** to configure Windows Defender use **Add-MpPreference** to extend settings such as Exclusion paths



```
PS C:\> Set-MpPreference -EnableNetworkProtection Enabled
PS C:\> Set-MpPreference -MAPSReporting Advanced
PS C:\> Set-MpPreference -PUAProtection Enabled
PS C:\> |
```

Managing the Microsoft Security stack with PowerShell – Windows Defender

Mpcmdrun.exe provides an option to verify exclusions, **validate-Defenderexclusions** is a wrapper script I created for the use with PowerShell <https://gist.github.com/alexverboon/b82348b4bce58092bef45c96d985a709>

The screenshot shows two windows side-by-side. The left window is a standard Windows Command Prompt (cmd) window titled 'cmd'. It contains a help message for the '-CheckExclusion' command:

```
-CheckExclusion -path <path>
    Checks whether <path> is excluded. It can be either a path, or a file.
```

The right window is a PowerShell window titled 'Administrator: Powershell'. It runs three separate commands using the 'Validate-DefenderExclusion' script:

```
PS C:\Dev\Alex\MSSecurityPowerShell\DefenderAV> . .\Validate-DefenderExclusions.ps1
PS C:\Dev\Alex\MSSecurityPowerShell\DefenderAV> Validate-DefenderExclusion -Path C:\AutomatedLab-VMs\
C:\AutomatedLab-VMs\ [\Device\HarddiskVolume3\AutomatedLab-VMs] is excluded. Exit code is 0.

Path          Excluded
----          -----
C:\AutomatedLab-VMs\ True

PS C:\Dev\Alex\MSSecurityPowerShell\DefenderAV> Validate-DefenderExclusion -Path C:\temp\demo\
C:\temp\demo\ [\Device\HarddiskVolume3\temp\demo] is excluded. Exit code is 0.

Path          Excluded
----          -----
C:\temp\demo\ True

PS C:\Dev\Alex\MSSecurityPowerShell\DefenderAV> Validate-DefenderExclusion -Path C:\temp\
C:\temp\ [\Device\HarddiskVolume3\temp] is not excluded. Exit code is 1.

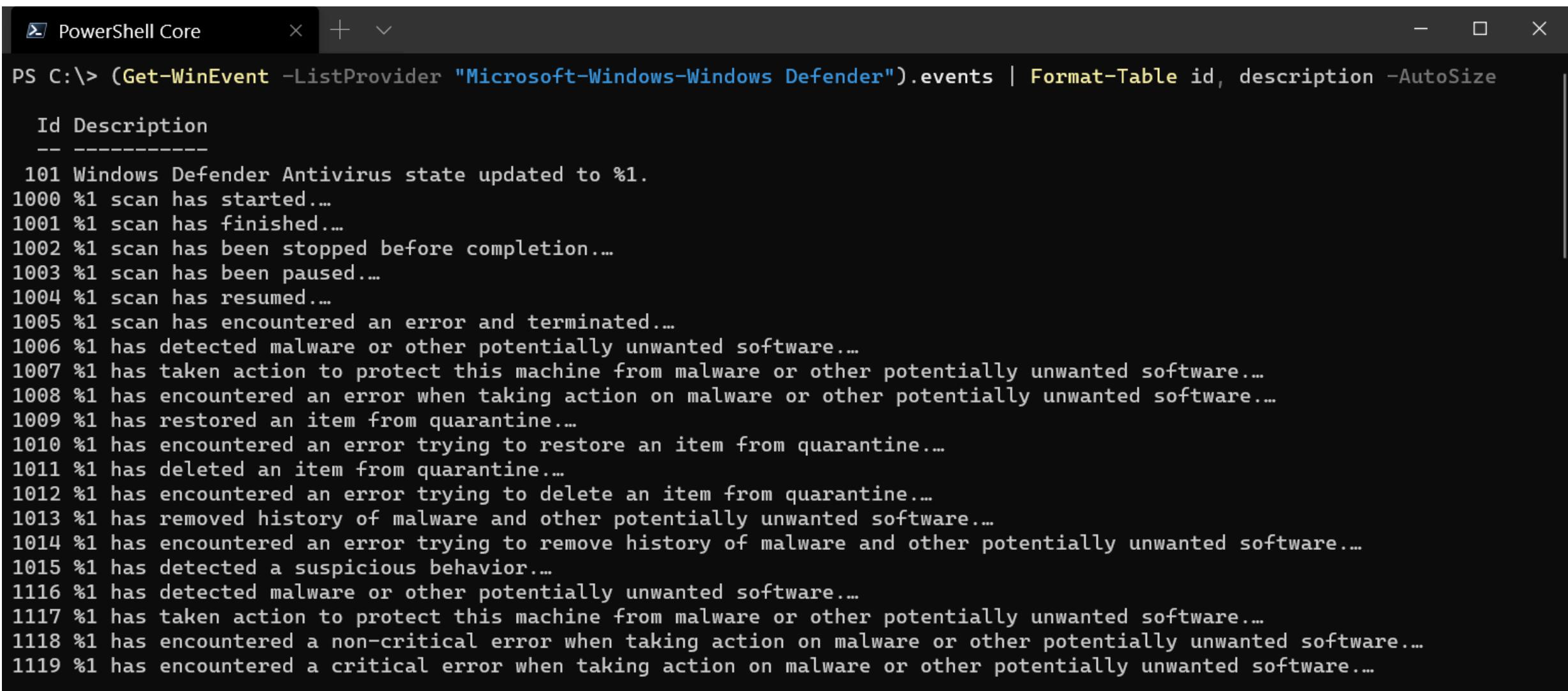
Path          Excluded
----          -----
C:\temp\ False

PS C:\Dev\Alex\MSSecurityPowerShell\DefenderAV> |
```

Managing the Microsoft Security stack with PowerShell – Windows Defender

Get a list of all possible Windows event IDs for Windows Defender

```
(Get-WinEvent -ListProvider "Microsoft-Windows-Windows Defender").Events | Format-Table id,Description -AutoSize
```



The screenshot shows a PowerShell Core window titled "PowerShell Core". The command `(Get-WinEvent -ListProvider "Microsoft-Windows-Windows Defender").events | Format-Table id, description -AutoSize` is run at the prompt. The output lists various Windows event IDs and their descriptions, such as 101 for antivirus state updates and 1000 for scan start notifications.

Id	Description
101	Windows Defender Antivirus state updated to %1.
1000	%1 scan has started....
1001	%1 scan has finished....
1002	%1 scan has been stopped before completion....
1003	%1 scan has been paused....
1004	%1 scan has resumed....
1005	%1 scan has encountered an error and terminated....
1006	%1 has detected malware or other potentially unwanted software....
1007	%1 has taken action to protect this machine from malware or other potentially unwanted software....
1008	%1 has encountered an error when taking action on malware or other potentially unwanted software....
1009	%1 has restored an item from quarantine....
1010	%1 has encountered an error trying to restore an item from quarantine....
1011	%1 has deleted an item from quarantine....
1012	%1 has encountered an error trying to delete an item from quarantine....
1013	%1 has removed history of malware and other potentially unwanted software....
1014	%1 has encountered an error trying to remove history of malware and other potentially unwanted software....
1015	%1 has detected a suspicious behavior....
1116	%1 has detected malware or other potentially unwanted software....
1117	%1 has taken action to protect this machine from malware or other potentially unwanted software....
1118	%1 has encountered a non-critical error when taking action on malware or other potentially unwanted software....
1119	%1 has encountered a critical error when taking action on malware or other potentially unwanted software....

Managing the Microsoft Security stack with PowerShell – Windows Defender

Find specific Event IDs

```
PS C:\>
PS C:\> $EventFilter = @{
>>     ID = 1000,1001
>>     ProviderName = "Microsoft-Windows-Windows Defender"
>> }
PS C:\> Get-WinEvent -FilterHashtable $EventFilter | format-table -autosize

ProviderName: Microsoft-Windows-Windows Defender

TimeCreated           Id LevelDisplayName Message
-----              -- -----
7/23/2020 5:21:13 AM 1001 Information      Windows Defender Antivirus scan has finished...
7/23/2020 5:18:41 AM 1000 Information      Windows Defender Antivirus scan has started...

PS C:\>
```

Managing the Microsoft Security stack with PowerShell – Windows Defender – Exploit Guard Events

Windows Defender Exploit Guard Network Protection, Attack Surface Reduction and Controlled Folder Access events are written to the Windows Event log: Microsoft-Windows-Windows Defender

#Controlled Folder Access

1124 Audit

1123 Block

Network Protection

1125 Audit

1126 Block

Attack Surface Rules

1121 Block

1122 Audit

The **Get-DefenderEEvents** cmdlet allows you to quickly pull those events from the Windows Event log

<https://www.verboon.info/2019/05/retrieving-windows-defender-exploit-guard-windows-event-logs-with-powershell/>

Managing the Microsoft Security stack with PowerShell – Windows Defender – Attack Surface Reduction

<https://github.com/sassdawe/HardenWindows/tree/main/WindowsDefender/DefenderASR>

```
Administrator: PowerShell      X + V      - □ ×

PS C:\Users\Administrator> get-command -module defenderasr

 CommandType      Name          Version   Source
-----      ----          -----   -----
 Alias        List-DAsrRules    0.0.4     defenderasr
 Function      Backup-DefenderAsrSetting 0.0.4     defenderasr
 Function      Get-DefenderAsrRule    0.0.4     defenderasr
 Function      Show-DefenderAsrRule   0.0.4     defenderasr

PS C:\Users\Administrator> List-DAsrRules

ID Name
-- --
0 Block executable content from email client and webmail
1 Block all Office applications from creating child processes
2 Block Office applications from creating executable content
3 Block Office applications from injecting code into other processes
4 Block JavaScript or VBScript from launching downloaded executable content
5 Block execution of potentially obfuscated scripts
6 Block Win32 API calls from Office macro
7 Block executable files from running unless they meet a prevalence, age, or trusted list criterion
8 Use advanced protection against ransomware
9 Block credential stealing from the Windows local security authority subsystem (lsass.exe)
10 Block process creations originating from PSEexec and WMI commands
11 Block untrusted and unsigned processes that run from USB
12 Block Office communication application from creating child processes
13 Block Adobe Reader from creating child processes
14 Block persistence through WMI event subscription

GUID
-----
be9ba2d9-53ea-4cd... d4f940ab-401b-4ef...
3b576869-a4ec-452... 75668c1f-73b5-4cf...
d3e037e1-3eb8-44c... 5beb7efe-fd9a-455...
92e97fa1-2edf-447... 01443614-cd74-433...
c1db55ab-c21a-463... 9e6c4e1f-7d60-472...
b1e49aac-8f56-428... b2b3f03d-6a65-4f7...
26190899-1602-49e... 7674ba52-37eb-4a4...
e6db77e5-3df2-4cf...

PS C:\Users\Administrator>
```

Managing the Microsoft Security stack with PowerShell – Windows Defender – Attack Surface Reduction

https://github.com/anthonws/MDatP_PoSh_Scripts

The screenshot shows a dual-pane interface. On the left is a dark-themed PowerShell window titled "Administrator: PowerShell" with the command ".\ASR_Rules_PoSh_GUI.ps1" running. On the right is a light-themed "ASR Rules PoSh GUI" window titled "Attack Surface Reduction Rules". The GUI lists 15 rules, each with three radio button options: Disabled, Audit (selected), and Enabled. At the bottom of the GUI, it shows "Total Numbers: 0 15 0" and has "Save Changes" and "Reset" buttons.

Rule Description	Disabled	Audit	Enabled
Block all Office applications from creating child processes	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Block execution of potentially obfuscated scripts	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Block Win32 API calls from Office macro	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Block Office applications from creating executable content	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Block Office applications from injecting code into other processes	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Block Adobe Reader from creating child processes	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Block Win32 API calls from Office macro	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Block credential stealing from the Windows local security authority subsystem (lsass.exe)	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Block untrusted and unsigned processes that run from USB	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Block executable content from email client and webmail	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Block executable files from running unless they meet a prevalence, age, or trusted list criterion	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Block process creations originating from PSEexec	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Block JavaScript or VBScript from launching downloaded executable content	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Block Adobe Reader from creating child processes	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Block persistence through WMI event subscription	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Total Numbers:	0	15	0

Managing the Microsoft Security stack with PowerShell – Windows Defender – Network Protection Events

Windows PowerShell PowerShell Core

```
PS C:\dev\alex\MSSecurityPowerShell\DefenderASR> Invoke-RestMethod -Uri "https://smartscreentestratings2.net"
Invoke-RestMethod : The request was aborted: Could not create SSL/TLS secure channel.
At line:1 char:1
+ Invoke-RestMethod -Uri "https://smartscreentestratings2.net"
+ ~~~~~~
+ CategoryInfo          : InvalidOperation: (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-RestMethod]
+ FullyQualifiedErrorMessage : WebCmdletWebResponseException,Microsoft.PowerShell.Commands.Invoke-RestMethodCommand
PS C:\dev\alex\MSSecurityPowerShell\DefenderASR> |
```

Windows Security

This content is blocked

For your protection, your administrator is not allowing you to access content from smartscreentestratings2.net.

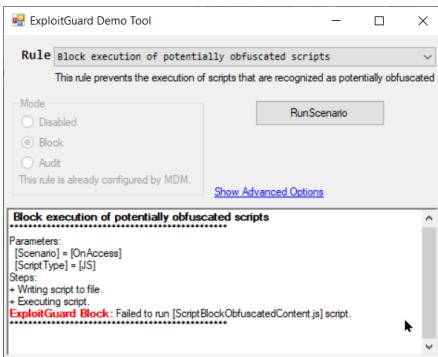
OK Unblock

Windows PowerShell PowerShell Core

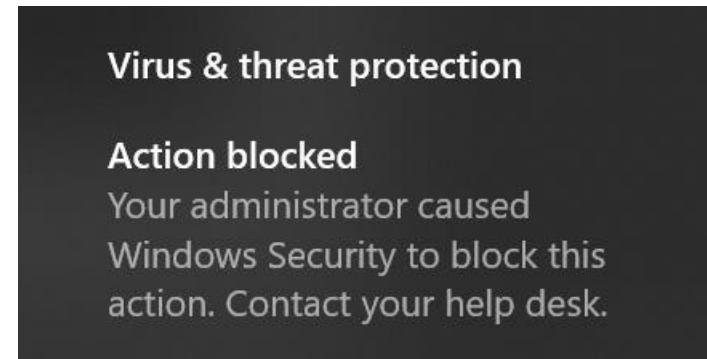
```
PS C:\Dev\Alex\MSSecurityPowerShell\DefenderASR> Get-DefenderEGEvents -Component NP -EGMode Block | fl
```

TimeCreated	ProviderName	Id	Message
8/25/2020 4:05:49 PM	Microsoft-Windows-Windows Defender	1126	Your IT administrator has caused Microsoft Defender Exploit Guard to block a potentially dangerous network connection. Detection time: 2020-08-25T14:05:49.126Z User: S-1-12-1-2321900437-1234520018-2326948268-2308314766 Destination: https://smartscreentestratings2.net Process Name: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

Managing the Microsoft Security stack with PowerShell – Windows Defender – Attack Surface Protection Events



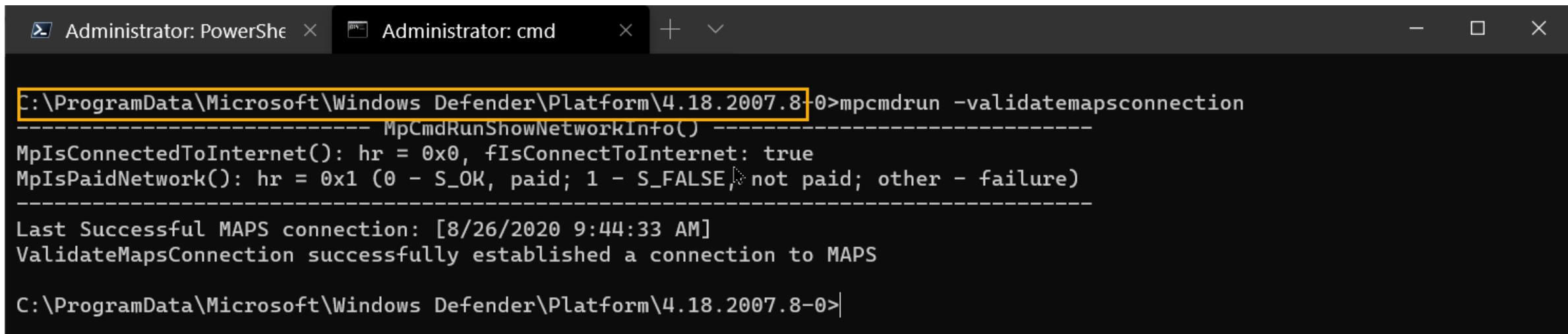
Executing obfuscated
scripts



```
Windows PowerShell    ×   PowerShell Core    ×   +   -   ×
PS C:\Dev\Alex\MSSecurityPowerShell\DefenderASR> Get-DefenderEGEvents -Component ASR -EGMode Block | fl
TimeCreated : 8/25/2020 4:09:25 PM
ProviderName : Microsoft-Windows-Defender
Id          : 1121
Message      : Microsoft Defender Exploit Guard has blocked an operation that is not allowed by your IT
               administrator.
               For more information please contact your IT administrator.
               ID: 5BEB7EFE-FD9A-4556-801D-275E5FFC04CC
               Detection time: 2020-08-25T14:09:25.304Z
               User: AzureAD\AlexVerboon
               Path: C:\ProgramData\AntiMalwareTest\8_25_2020\846cad1-dcf1-41e1-b409-519bf5d37d2c\Script
Block0
               bfuscatingContent-{c89e36a0-5606-43ee-b30b-72e8214b35ee}.js
               Process Name: C:\Windows\System32\wscript.exe
               Security intelligence Version: 1.321.2158.0
               Engine Version: 1.1.17300.4
               Product Version: 4.18.2007.8
```

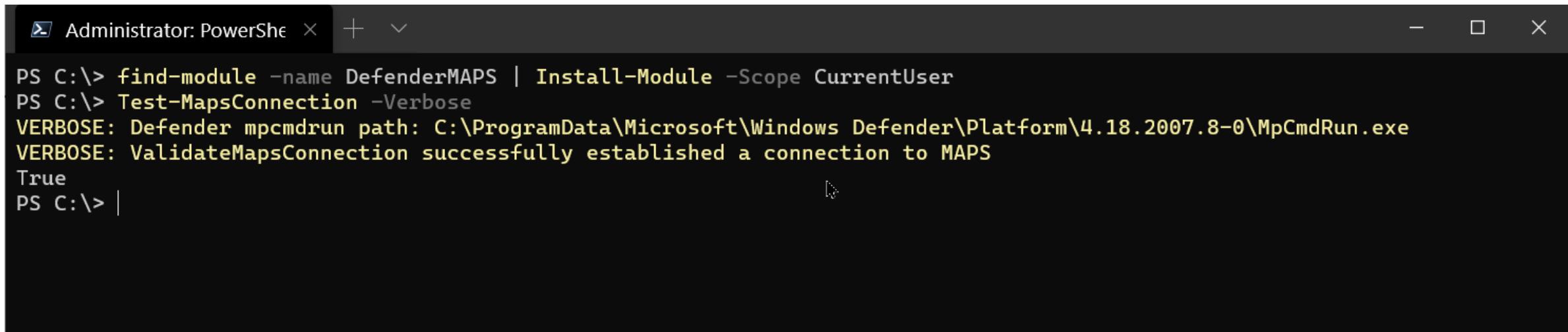
Managing the Microsoft Security stack with PowerShell – Windows Defender – MAPS

Test Windows Defender MAPS Connection



```
C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.2007.8->mpcmdrun -validatemapsconnection
----- MpCmdRunShowNetworkInfo() -----
MpIsConnectedToInternet(): hr = 0x0, fIsConnectToInternet: true
MpIsPaidNetwork(): hr = 0x1 (0 - S_OK, paid; 1 - S_FALSE, not paid; other - failure)
-----
Last Successful MAPS connection: [8/26/2020 9:44:33 AM]
ValidateMapsConnection successfully established a connection to MAPS

C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.2007.8->
```



```
PS C:\> find-module -name DefenderMAPS | Install-Module -Scope CurrentUser
PS C:\> Test-MapsConnection -Verbose
VERBOSE: Defender mpcmdrun path: C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.2007.8-0\MpCmdRun.exe
VERBOSE: ValidateMapsConnection successfully established a connection to MAPS
True
PS C:\> |
```



PSMDATP – PowerShell Module for Microsoft Defender
Advanced Threat Protection
<https://github.com/alexverboon/PSMDATP>

Microsoft Defender Advanced Threat Protection

prevent, detect, investigate, and respond to advanced threats.



Threat &
Vulnerability
Management



Attack
surface
reduction



Next-
generation
protection



Endpoint
detection and
response



Automated
investigation and
remediation



Microsoft
Threat
Experts

Centralized configuration and administration, APIs

Microsoft Threat Protection

Managing the Microsoft Security stack with PowerShell – Defender ATP

```
Administrator: PowerShell + X
PS C:\> find-module -name PSMDATP

Version          Name          Repository          Description
----          Name          Repository          Description
-----          -----          -----          -----
1.0.0          PSMDATP          PSGallery          "Manage Microsoft Defender ATP with Powe...
PS C:\> find-module -name PSMDATP | Install-Module -Scope CurrentUser
PS C:\> |
```

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

API / Permissions name	Type	Description	Admin consent req...	Status
User.Read	Delegated	Sign in and read user profile	-	
Azure Active Directory Graph (1)				
AdvancedQuery.ReadAll	Application	Run advanced queries	Yes	
Alert.Read.All	Application	Read all alerts	Yes	
Alert.ReadWrite.All	Application	Read and write all alerts	Yes	
Machine.CollectForensics	Application	Collect forensics	Yes	
Machine.Isolate	Application	Isolate machine	Yes	
Machine.Offboard	Application	Offboard machine	Yes	
Machine.Read.All	Application	Read all machine profiles	Yes	
Machine.ReadWrite.All	Application	Read and write all machine information	Yes	
Machine.RestrictExecution	Application	Restrict code execution	Yes	
Machine.Scan	Application	Scan machine	Yes	
Score.Read.All	Application	Read Threat and Vulnerability Management score	Yes	
SecurityConfiguration.Read.All	Application	Read Threat and Vulnerability Management security co...	Yes	
SecurityRecommendation.Rea...	Application	Read Threat and Vulnerability Management security re...	Yes	
Software.Read.All	Application	Read Threat and Vulnerability Management software in...	Yes	
Ti.ReadWrite	Application	Read and write IOCs belonging to the app	Yes	
Ti.ReadWrite.All	Application	Read and write all IOCs	Yes	
Vulnerability.Read.All	Application	Read Threat and Vulnerability Management vulnerabilit...	Yes	

PoshMTPconfig.json - Notepad

File Edit Format View Help

```
"API_MDATP": {  
    "AppName": "MDATPOPS",  
    "OAuthUri": "https://login.windows.net/[REDACTED]/oauth2/token",  
    "ClientID": "[REDACTED]",  
    "ClientSecret": "[REDACTED]"  
},  
"API_MSGRAPH": {  
    "AppName": "xMSGraph",  
    "OAuthUri": "https://login.windows.net/[REDACTED]/oauth2/token",  
    "ClientID": "[REDACTED]",  
    "ClientSecret": "[REDACTED]"  
}  
}
```

Granted for Contoso ...
Granted for Contoso ...

Managing the Microsoft Security stack with PowerShell – Defender ATP

```
Administrator: PowerShell + ▾ - ▾ X

PS C:\> Get-MDATPAlert | Select-object AlertCreationTime, Title, Severity
VERBOSE: GET https://api.securitycenter.windows.com/api/alerts?sinceTimeUtc=2020-08-26T08%3A10%3A06.3191062Z with 0-byte
payload
VERBOSE: received 161258-byte response of content type application/json
VERBOSE: Content encoding: utf-8

alertCreationTime      title                                     severity
-----      -----
8/25/2020 2:22:22 PM  firefox usage detected                Low
8/10/2020 12:04:21 PM  firefox usage detected                Low
8/25/2020 2:02:28 PM  Suspicious connection blocked by network protection  Informational
8/20/2020 5:39:08 AM  'SoreFang' malware was prevented    Informational
8/20/2020 5:39:08 AM  'CryptInject' malware was prevented   Informational
8/20/2020 5:39:10 AM  'Emotet' malware was prevented       Informational
8/19/2020 8:00:50 PM  'Emotet' malware was detected        Informational
8/19/2020 8:03:53 PM  'CryptInject' malware was detected   Informational
8/19/2020 8:08:14 PM  'SoreFang' malware was detected      Informational
8/19/2020 7:44:24 PM  'Wacatac' malware was detected       Informational
8/19/2020 7:55:25 PM  An active 'Wacatac' malware was blocked  Low
8/19/2020 8:01:26 PM  An active 'Emotet' malware was blocked  Low
8/19/2020 7:54:50 PM  Anomaly detected in ASEP registry     Medium
8/19/2020 8:03:05 PM  Suspicious file dropped             Medium
8/19/2020 8:33:43 PM  Suspicious file registered as a service  Medium
```

Managing the Microsoft Security stack with PowerShell – Defender ATP

```
Administrator: PowerShell + ▾  
PS C:\> Get-MDATPDevice -DeviceName lclient02  
VERBOSE: GET https://api.securitycenter.windows.com/api/machines?$filter=ComputerDNSName eq 'lclient02' with 0-byte payload  
VERBOSE: received 667-byte response of content type application/json  
VERBOSE: Content encoding: utf-8  
  
id : 57b63cb49223ce246f26d57a81287c9f17893bea  
computerDnsName : lclient02  
firstSeen : 7/10/2020 6:00:22 PM  
lastSeen : 8/26/2020 7:34:05 AM  
osPlatform : Windows10  
osVersion :  
osProcessor : x64  
version : 2004  
lastIpAddress : 172.18.192.1  
lastExternalIpAddress : 77.56.162.123  
agentVersion : 10.7430.19041.423  
osBuild : 19041  
healthStatus : Active  
deviceValue : Normal  
rbacGroupId : 913  
rbacGroupName : All Devices  
riskScore : High  
exposureLevel : Medium  
aadDeviceId : 9eb5a403-e50a-435a-ae25-32cc8510ad6a  
machineTags : {}  
  
PS C:\> |
```

Microsoft Defender Security Center

Devices > lclient02

lclient02 High Active

Tags: Data sensitivity:Top Secret

Security Info:

- Open incidents: 17
- Active alerts: 53
- Exposure level: Medium
- Risk level: High

Azure ATP alerts: Device not found in Azure ATP

Device details:

- Domain: AAD joined
- OS: Windows 10 x64, Version 2004, Build 19041.450

Asset group

Managing the Microsoft Security stack with PowerShell – Defender ATP

Isolating the device from the network

The screenshot illustrates the process of isolating a device using PowerShell and observing the resulting state in the Windows Firewall & network protection interface and the Microsoft Defender ATP Action center.

PowerShell Command:

```
PS C:\> Start-MDATPIsolation -DeviceName lclient02 -IsolationType Full -Comment "isolating device as per incident INC039393"
```

Verbose Output:

```
VERBOSE: GET https://api.securitycenter.windows.com/api/machines with 0-byte payload  
VERBOSE: received 2325-byte response of content type application/json  
VERBOSE: Content encoding: utf-8  
09ee26a3-ec30-4fec-8910-0e4fd8867187  
PS C:\>
```

Windows Firewall & network protection:

Network Disabled
Your administrator has caused Microsoft Defender to disconnect your device.
Contact your help desk.

Action center (Microsoft Defender ATP):

Investigation package collection

Submission time	Status
Aug 25, 2020, 4:13:45 PM	↓ Package collection package available

Package collection submitted by oa@verboon.online on Aug 25, 2020, 4:13:45 PM
Collect investigation package was triggered by hunting bulk action

App restriction

Submission time	Status
Aug 10, 2020, 11:58:53 AM	App restriction removal removed

Code execution restriction submitted by oa@verboon.online on Aug 10, 2020, 11:58:53 AM
xczf

Device isolation

Submission time	Status
Aug 26, 2020, 10:16:19 AM	Device isolation configuration applied

Device isolation submitted by 2602c190-202a-4217-b647-e70e4c9f157e on Aug 26, 2020, 10:16:19 AM
isolating device as per incident INC039393

Managing the Microsoft Security stack with PowerShell – Defender ATP

```
Windows PowerShell  PowerShell Core + - ×
PS C:\Users\AlexVerboon> Get-MDATPAlert -Severity High
VERBOSE: GET https://api.securitycenter.windows.com/api/alerts?sinceTimeUtc=2020-09-09T18%3A41%3A45.6929104Z with 0-byte
payload
VERBOSE: received 199002-byte response of content type application/json
VERBOSE: Content encoding: utf-8

id : da637329116998998878_-1447647443
incidentId : 47
investigationId :
assignedTo :
severity : High
status : New
classification :
determination :
investigationState : UnsupportedAlertType
detectionSource : WindowsDefenderAtp
category : CredentialAccess
threatFamilyName :
title : Sensitive credential memory read
description : A process scanned or dumped memory from the Local Security Authority Subsystem Service
(lsass.exe). Accessing this process memory allows the attacker to extract secrets such as
authentication hashes or passwords. A copy of this memory may be written to the file system and
exfiltrated to extract these credentials offline.
alertCreationTime : 8/13/2020 10:34:59 AM
firstEventTime : 8/13/2020 10:11:34 AM
lastEventTime : 8/13/2020 10:11:34 AM
lastUpdateTime : 8/13/2020 10:35:01 AM
resolvedTime :
machineId : 57b63cb49223ce246f26d57a81287c9f17893bea
```

Managing the Microsoft Security stack with PowerShell – Defender ATP

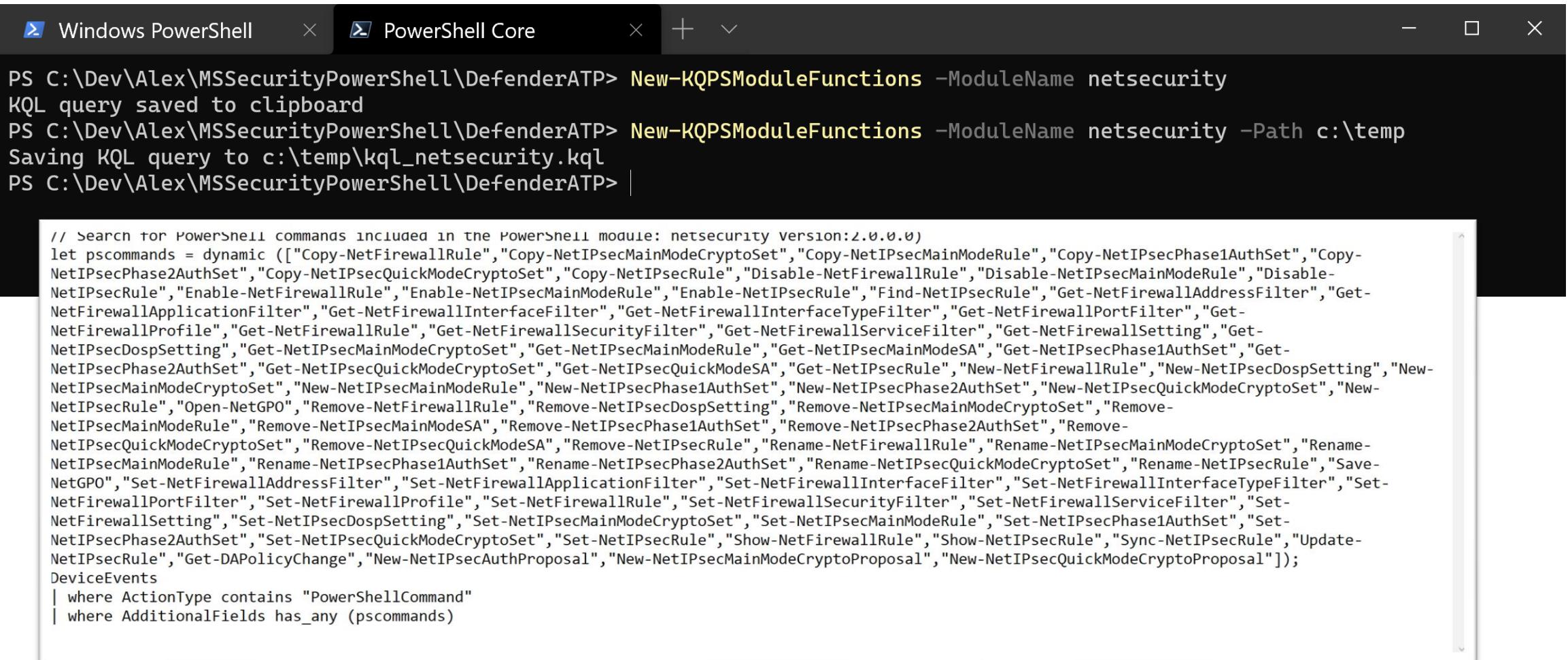
```
Windows PowerShell  ×  PowerShell Core  ×  +  ↻  -  □  ×  
PS C:\Users\AlexVerboon> Start-MDATPAVScan -DeviceName lclient02 -ScanType Quick  
VERBOSE: GET https://api.securitycenter.windows.com/api/machines with 0-byte payload  
VERBOSE: received 4090-byte response of content type application/json  
VERBOSE: Content encoding: utf-8  
0be12ca5-fdf8-440d-a40e-77e7afe62d1c  
PS C:\Users\AlexVerboon> Get-MDATPDeviceAction -DeviceName lclient02  
VERBOSE: GET https://api.securitycenter.windows.com/api/machines?$filter=ComputerDNSName eq 'lclient02' with 0-byte payload  
VERBOSE: received 669-byte response of content type application/json  
VERBOSE: Content encoding: utf-8  
VERBOSE: GET https://api.securitycenter.windows.com/api/machines/57b63cb49223ce246f26d57a81287c9f17893bea with 0-byte payload  
VERBOSE: received 665-byte response of content type application/json  
VERBOSE: Content encoding: utf-8  
VERBOSE: GET https://api-us.securitycenter.windows.com/api/machineactions/?$filter=machineId+eq+'57b63cb49223ce246f26d57a81287c9f17893bea' with 0-byte payload  
VERBOSE: received 23970-byte response of content type application/json  
VERBOSE: Content encoding: utf-8  
  
id : 0be12ca5-fdf8-440d-a40e-77e7afe62d1c  
type : RunAntiVirusScan  
requestor : 2602c190-202a-4217-b647-e70e4c9f157e  
requestorComment : submitted by automation  
status : Pending  
machineId : 57b63cb49223ce246f26d57a81287c9f17893bea  
computerDnsName : lclient02  
creationDateTimeUtc : 9/9/2020 6:50:40 PM  
lastUpdateDateTimeUtc : 9/9/2020 6:50:40 PM  
cancellationRequestor :
```

Managing the Microsoft Security stack with PowerShell – Defender ATP

```
Windows PowerShell  PowerShell Core + < > PS C:\Users\AlexVerboon> Get-MDATPIndicator -IndicatorType DomainName  
VERBOSE: GET https://api.securitycenter.windows.com/api/indicators?client_id=2602c190-202a-4217-b647-e70e4c9f157e&grant_type=client_credentials&redirectUri=https%3A%2F%2Flocalhost%3A8000&resource=https%3A%2F%2Fapi.securitycenter.windows.com&client_secret=576af19b-6d85-44de-bf67-8286a3bf4f49 with 0-byte payload  
VERBOSE: received 32718-byte response of content type application/json  
VERBOSE: Content encoding: utf-8  
  
id : 68  
indicatorValue : godaddy.com  
indicatorType : DomainName  
action : AlertAndBlock  
createdBy : ca7154ab-7eab-4bf1-ba32-f2e119637859  
source : Microsoft Cloud App Security  
sourceType : AadApp  
severity : Informational  
category : 1  
application : GoDaddy  
educateUrl :  
bypassDurationHours :  
title : Connection to a blocked cloud application was detected  
description : Endpoint had established a connection with a risky cloud application godaddy.com. This connection was classified as risky according to your organization Microsoft Cloud App Security administrator. You can view the respective indicator under the URLs/Domain tab or from within the Microsoft Cloud App Security portal.  
recommendedActions :  
creationTimeDateTimeUtc : 9/2/2020 5:44:19 PM  
expirationTime :  
lastUpdateTime : 9/2/2020 5:44:19 PM  
lastUpdatedBy :
```

Managing the Microsoft Security stack with PowerShell – Defender ATP – Advanced Hunting

User PowerShell to generate a KQL query that looks for executed cmdlets included in a PowerShell Module



The screenshot shows a Windows PowerShell window with two tabs: "Windows PowerShell" and "PowerShell Core". The "Windows PowerShell" tab is active, displaying the following command and its output:

```
PS C:\Dev\Alex\MSSecurityPowerShell\DefenderATP> New-KQPSModuleFunctions -ModuleName netsecurity
KQL query saved to clipboard
PS C:\Dev\Alex\MSSecurityPowerShell\DefenderATP> New-KQPSModuleFunctions -ModuleName netsecurity -Path c:\temp
Saving KQL query to c:\temp\kql_netsecurity.kql
PS C:\Dev\Alex\MSSecurityPowerShell\DefenderATP>
```

Below the command history, a large block of KQL code is displayed in a scrollable pane. The code is a search query for PowerShell commands included in the netsecurity module, version 2.0.0.0. It lists numerous cmdlets such as Copy-NetFirewallRule, Copy-NetIPsecMainModeCryptoSet, Copy-NetIPsecMainModeRule, Copy-NetIPsecPhase1AuthSet, Copy-NetIPsecPhase2AuthSet, Copy-NetIPsecQuickModeCryptoSet, Copy-NetIPsecRule, Disable-NetFirewallRule, Disable-NetIPsecMainModeRule, Disable-NetIPsecRule, Enable-NetFirewallRule, Enable-NetIPsecMainModeRule, Enable-NetIPsecRule, Find-NetIPsecRule, Get-NetFirewallAddressFilter, Get-NetFirewallApplicationFilter, Get-NetFirewallInterfaceFilter, Get-NetFirewallInterfaceTypeFilter, Get-NetFirewallPortFilter, Get-NetFirewallProfile, Get-NetFirewallRule, Get-NetFirewallSecurityFilter, Get-NetFirewallServiceFilter, Get-NetFirewallSetting, Get-NetIPsecDospSetting, Get-NetIPsecMainModeCryptoSet, Get-NetIPsecMainModeRule, Get-NetIPsecMainModeSA, Get-NetIPsecPhase1AuthSet, Get-NetIPsecPhase2AuthSet, Get-NetIPsecQuickModeCryptoSet, Get-NetIPsecQuickModeSA, Get-NetIPsecRule, New-NetFirewallRule, New-NetIPsecDospSetting, New-NetIPsecMainModeCryptoSet, New-NetIPsecMainModeRule, New-NetIPsecPhase1AuthSet, New-NetIPsecPhase2AuthSet, New-NetIPsecQuickModeCryptoSet, New-NetIPsecRule, Open-NetGPO, Remove-NetFirewallRule, Remove-NetIPsecDospSetting, Remove-NetIPsecMainModeCryptoSet, Remove-NetIPsecMainModeRule, Remove-NetIPsecMainModeSA, Remove-NetIPsecPhase1AuthSet, Remove-NetIPsecPhase2AuthSet, Remove-NetIPsecQuickModeCryptoSet, Remove-NetIPsecQuickModeSA, Remove-NetIPsecRule, Rename-NetFirewallRule, Rename-NetIPsecMainModeCryptoSet, Rename-NetIPsecMainModeRule, Rename-NetIPsecPhase1AuthSet, Rename-NetIPsecPhase2AuthSet, Rename-NetIPsecQuickModeCryptoSet, Rename-NetIPsecRule, Save-NetGPO, Set-NetFirewallAddressFilter, Set-NetFirewallApplicationFilter, Set-NetFirewallInterfaceFilter, Set-NetFirewallInterfaceTypeFilter, Set-NetFirewallPortFilter, Set-NetFirewallProfile, Set-NetFirewallRule, Set-NetFirewallSecurityFilter, Set-NetFirewallServiceFilter, Set-NetFirewallSetting, Set-NetIPsecDospSetting, Set-NetIPsecMainModeCryptoSet, Set-NetIPsecMainModeRule, Set-NetIPsecPhase1AuthSet, Set-NetIPsecPhase2AuthSet, Set-NetIPsecQuickModeCryptoSet, Set-NetIPsecRule, Show-NetFirewallRule, Show-NetIPsecRule, Sync-NetIPsecRule, Update-NetIPsecRule, Get-DAPolicyChange, New-NetIPsecAuthProposal, New-NetIPsecMainModeCryptoProposal, New-NetIPsecQuickModeCryptoProposal];

DeviceEvents
| where ActionType contains "PowerShellCommand"
| where AdditionalFields has_any (pscommands)

<https://gist.github.com/alexverboon/9ccf8af7569103397da2b8ba4079529d>



AzureAD

Multifactor Authentication

Who has registered for MFA and who hasn't?

What MFA methods are registered?

Script for Azure MFA authentication method analysis

<https://github.com/Azure-Samples/azure-mfa-authentication-method-analysis/tree/master/>

Managing the Microsoft Security stack with PowerShell – AzureAD Multifactor Authentication

The screenshot shows the Azure AD portal interface with a sidebar on the left containing icons for Overview, Getting started, Diagnose and solve problems, Manage (selected), Users, and Groups. The main area displays 'tenant information' with sections for Your role (Global reader), License (Azure AD Premium P2), and Tenant ID. A blue arrow points from the Tenant ID field in the portal to a PowerShell command in the foreground.

Windows PowerShell | **PowerShell Core**

```
PS C:\Dev\Alex\MSSecurityPowerShell\AzureAD\MFA> $mfareginfo = .\MfaAuthMethodsAnalysis.ps1 -TenantId [REDACTED]  
PS C:\Dev\Alex\MSSecurityPowerShell\AzureAD\MFA> $mfareginfo | gm
```

TypeName: System.Management.Automation.PSCustomObject

Name	MemberType	Definition
Equals	Method	bool Equals(System.Object obj)
GetHashCode	Method	int GetHashCode()
GetType	Method	type GetType()
ToString	Method	string ToString()
AltPhone	NoteProperty	object AltPhone=null
AppNotification	NoteProperty	object AppNotification=null
DefaultMethod	NoteProperty	object DefaultMethod=null
DisplayName	NoteProperty	string DisplayName=[REDACTED]
MfaAuthMethodCount	NoteProperty	int MfaAuthMethodCount=0
OathTotp	NoteProperty	object OathTotp=null
ObjectId	NoteProperty	string ObjectId=[REDACTED]
Phone	NoteProperty	object Phone=null
Recommendations	NoteProperty	Object[] Recommendations=System.Object[]
Sms	NoteProperty	object Sms=null
UserPrincipalName	NoteProperty	string UserPrincipalName=[REDACTED]

```
PS C:\Dev\Alex\MSSecurityPowerShell\AzureAD\MFA> |
```

Managing the Microsoft Security stack with PowerShell – AzureAD Multifactor Authentication

Get an overview of the MFA registration status

```
Windows PowerShell x PowerShell Core x | + - □ ×
PS C:\Dev\Alex\MSSecurityPowerShell\AzureAD\MFA> $no_mfa = $mfareginfo | where MfaAuthMethodCount -eq 0
PS C:\Dev\Alex\MSSecurityPowerShell\AzureAD\MFA> $no_mfa.count
2599
PS C:\Dev\Alex\MSSecurityPowerShell\AzureAD\MFA> $mfa = $mfareginfo | where MfaAuthMethodCount -gt 0
PS C:\Dev\Alex\MSSecurityPowerShell\AzureAD\MFA> $mfa.count
131
PS C:\Dev\Alex\MSSecurityPowerShell\AzureAD\MFA>
```

User details

```
Windows PowerShell x PowerShell Core x | + - □ ×
PS C:\Dev\Alex\MSSecurityPowerShell\AzureAD\MFA> $mfareginfo | where UserPrincipalName -like '*verale*'

UserPrincipalName : [REDACTED]
DisplayName       : Alex Verboon
ObjectId          : [REDACTED]
MfaAuthMethodCount : 4
DefaultMethod     : PhoneAppNotification
AppNotification   : Yes
OathTotp          : Yes
Sms               : Yes
Phone              : Yes
AltPhone           :
Recommendations   : {'Consider adding an alternative phone number for additional resilience.'}

PS C:\Dev\Alex\MSSecurityPowerShell\AzureAD\MFA>
```

Managing the Microsoft Security stack with PowerShell – AzureAD Multifactor Authentication

The screenshot shows the Azure AD portal interface. On the left, there's a sidebar with various management options like Properties, Members (Preview), Owners (Preview), etc. The main area displays a group named "MFA_ROLLOUT". The "Object Id" field is highlighted with a green border. A blue arrow points from this field to the "-TargetGroup" parameter in the PowerShell command below.

Windows PowerShell PowerShell Core

```
PS C:\Dev\Alex\MSSecurityPowerShell\AzureAD\MFA> $mfareginfo_group = .\MfaAuthMethodsAnalysis.ps1 -TenantId -TargetGroup
```

Windows PowerShell PowerShell Core

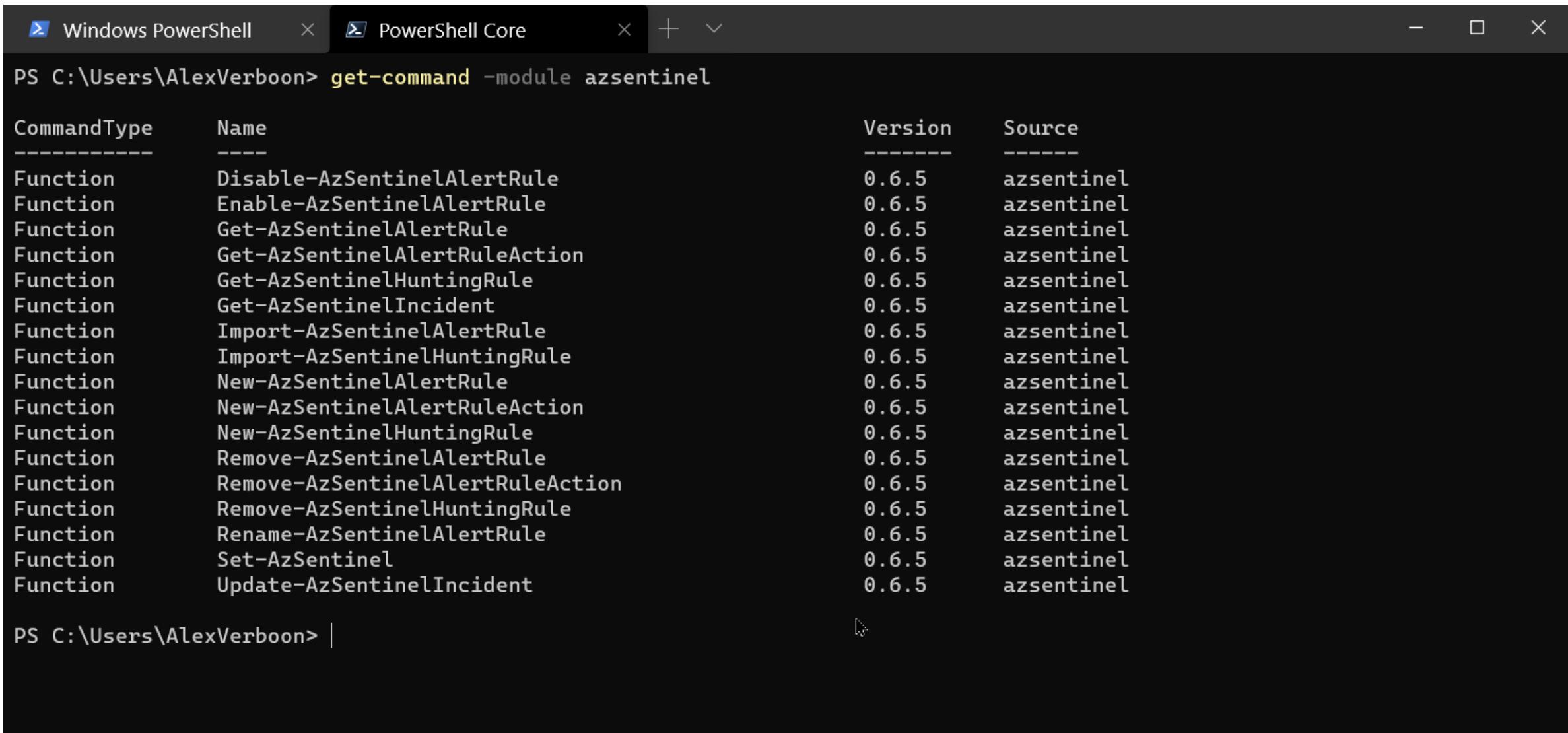
```
PS C:\Dev\Alex\MSSecurityPowerShell\AzureAD\MFA> $mfareginfo_group | where MfaAuthMethodCount -eq 0 | Select-object UserPrincipalName
```

UserPrincipalName

Azure Sentinel

Managing the Microsoft Security stack with PowerShell – Azure Sentinel

<https://github.com/wortell/AZSentinel>



The screenshot shows a Windows PowerShell window with two tabs: "Windows PowerShell" and "PowerShell Core". The "Windows PowerShell" tab is active and displays the command "get-command -module azsentinel". The output is a table listing 18 functions from the azsentinel module, all version 0.6.5 and sourced from azsentinel.

CommandType	Name	Version	Source
-----	-----	-----	-----
Function	Disable-AzSentinelAlertRule	0.6.5	azsentinel
Function	Enable-AzSentinelAlertRule	0.6.5	azsentinel
Function	Get-AzSentinelAlertRule	0.6.5	azsentinel
Function	Get-AzSentinelAlertRuleAction	0.6.5	azsentinel
Function	Get-AzSentinelHuntingRule	0.6.5	azsentinel
Function	Get-AzSentinelIncident	0.6.5	azsentinel
Function	Import-AzSentinelAlertRule	0.6.5	azsentinel
Function	Import-AzSentinelHuntingRule	0.6.5	azsentinel
Function	New-AzSentinelAlertRule	0.6.5	azsentinel
Function	New-AzSentinelAlertRuleAction	0.6.5	azsentinel
Function	New-AzSentinelHuntingRule	0.6.5	azsentinel
Function	Remove-AzSentinelAlertRule	0.6.5	azsentinel
Function	Remove-AzSentinelAlertRuleAction	0.6.5	azsentinel
Function	Remove-AzSentinelHuntingRule	0.6.5	azsentinel
Function	Rename-AzSentinelAlertRule	0.6.5	azsentinel
Function	Set-AzSentinel	0.6.5	azsentinel
Function	Update-AzSentinelIncident	0.6.5	azsentinel

PS C:\Users\AlexVerboon> |

Managing the Microsoft Security stack with PowerShell – Azure Sentinel

The screenshot shows a Visual Studio Code window with the title "hunting.json - Visual Studio Code". The left sidebar contains icons for search, file operations, and other development tools. The main editor area displays a JSON file named "hunting.json". The file content defines two hunting rules:

```
{}
  [
    {
      "analytics": [
        {
          "displayName": "HuntingRule01",
          "description": "test",
          "query": "SecurityEvent | where EventID == \"4688\" | where CommandLine contains \"-noni -ep bypass \$\"",
          "tactics": [
            "Persistence",
            "LateralMovement",
            "Collection"
          ]
        },
        {
          "displayName": "HuntingRule02",
          "description": "test",
          "query": "SecurityEvent | where EventID == \"4688\" | where CommandLine contains \"-noni -ep bypass \$\"",
          "tactics": [
            "Persistence",
            "LateralMovement"
          ]
        }
      ]
    }
}
```

The status bar at the bottom indicates the file is 1 line, 1 column long, with 2 spaces, in UTF-8 encoding, and is a JSON file. It also shows icons for navigation and search.

Managing the Microsoft Security stack with PowerShell – Azure Sentinel

```
Windows PowerShell  x  PowerShell Core  x  +  -  □  ×  
PS C:\Users\AlexVerboon> Import-AzSentinelHuntingRule -WorkspaceName MTPLabSentinel01 -SettingsFile "C:\Dev\Alex\MSSecurityPowerShell\sentinel\hunting.json"  
Started with Hunting rule: HuntingRule01  
Successfully created hunting rule: HuntingRule01 with status: OK  
  
Name : DisplayName  
Value : HuntingRule01  
  
Name : Category  
Value : Hunting Queries  
  
Name : Tags  
Value : {description, tactics, createdBy, createdTimeUtc}  
  
Name : Query  
Value : SecurityEvent | where EventID == "4688" | where CommandLine contains "-noni -ep bypass $"  
  
Started with Hunting rule: HuntingRule02  
Successfully created hunting rule: HuntingRule02 with status: OK  
  
Name : DisplayName  
Value : HuntingRule02
```

Managing the Microsoft Security stack with PowerShell – Azure Sentinel

Microsoft Azure Search resources, services, and docs (G+/)

Home > Azure Sentinel workspaces > Azure Sentinel

Azure Sentinel | Hunting

Selected workspace: 'mtplabsentinel01'

Search (Ctrl+ /) Refresh Last 24 hours New Query Run all queries Columns

94 Total queries 0 My bookmarks 0 Livestream Results MITRE ATT&CK™ LEARN MORE About hunting

Overview Logs News & guides

Threat management

Incidents Workbooks Hunting Notebooks (Preview) Entity behavior (Preview)

Configuration

Data connectors Analytics Playbooks Community Settings

Queries Livestream Bookmarks

Search queries Favorites : All Provider : All Data sources : All Tactics : All

Query	Provider	Data Source	Results	Tactics
Anomalous Login to Device	Microsoft	BehaviorAnalytics	--	Privilege Escalation
Anomalous Password Reset	Microsoft	AuditLogs +1 ⓘ	--	Impact
Anomalous RDP Activity	Microsoft	BehaviorAnalytics	--	Lateral Movement
Anomalous Resource Access	Microsoft	BehaviorAnalytics	--	Lateral Movement
Anomalous Role Assignment	Microsoft	AuditLogs +1 ⓘ	--	Persistence
Anomalous Sign-in Activity	Microsoft	SigninLogs +1 ⓘ	--	Persistence
HuntingRule01	Custom Queries	SecurityEvent	--	Exfiltration
HuntingRule02	Custom Queries	SecurityEvent	--	Exfiltration

< Previous 76 - 94 Next > Run Query View Results

No query selected Select a query to view more details

The screenshot shows the Azure Sentinel Hunting interface. On the left, there's a navigation sidebar with links like Overview, Logs, News & guides, Threat management (Incidents, Workbooks, Hunting), Configuration (Data connectors, Analytics, Playbooks, Community), and Settings. The 'Hunting' link under Threat management is highlighted. The main area has a search bar at the top. Below it, there are summary metrics: 94 Total queries, 0 My bookmarks, 0 Livestream Results, and a MITRE ATT&CK™ section with various counts. There are tabs for Queries, Livestream, and Bookmarks, with 'Queries' being active. A search bar for 'Search queries' is present. Below that are four filter buttons: Favorites : All, Provider : All, Data sources : All, and Tactics : All. The main content area is a table listing anomalies and hunting rules. The table has columns for Query, Provider, Data Source, Results, and Tactics. Rows include 'Anomalous Login to Device' (Microsoft, BehaviorAnalytics, --, Privilege Escalation), 'Anomalous Password Reset' (Microsoft, AuditLogs +1 ⓘ, --, Impact), 'Anomalous RDP Activity' (Microsoft, BehaviorAnalytics, --, Lateral Movement), 'Anomalous Resource Access' (Microsoft, BehaviorAnalytics, --, Lateral Movement), 'Anomalous Role Assignment' (Microsoft, AuditLogs +1 ⓘ, --, Persistence), 'Anomalous Sign-in Activity' (Microsoft, SigninLogs +1 ⓘ, --, Persistence), 'HuntingRule01' (Custom Queries, SecurityEvent, --, Exfiltration), and 'HuntingRule02' (Custom Queries, SecurityEvent, --, Exfiltration). A yellow box highlights the last three rows. At the bottom, there are navigation buttons for '< Previous' and 'Next >', and two action buttons 'Run Query' and 'View Results'. A message on the right says 'No query selected' and 'Select a query to view more details'.

Microsoft Graph

Managing the Microsoft Security stack with PowerShell – Microsoft Graph

```
Windows PowerShell  x  PowerShell Core  x  +  -  X
PS C:\> Get-MgSecurityAlert
Get-MgSecurityAlert_List: Auth token does not contain valid permissions or user does not have valid roles.
PS C:\> connect-graph -Scopes "SecurityEvents.Read.All", "SecurityEvents.ReadWrite.All" -TenantId
3-2e24ac9a9743
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code DGLXQBVNN to authenticate.
Welcome To Microsoft Graph!
PS C:\> Get-MgSecurityAlert

Id                               ActivityGroupName AssignedTo
--                               -----
a046a207-724b-9a5e-1b38-dc44699fcbee
da637351427275281792_-355679477
da637351553421775235_1682487945
da637351538674916243_1603903502
da637351539434516869_-937687580
                                         Automation
                                         Automation
                                         Automation
```

Managing the Microsoft Security stack with PowerShell – Microsoft Graph

Microsoft | Docs Documentation Learn Q&A Code Samples

Search Sign in

Microsoft Graph Guides API Reference Resources Developer Program

Getting started Graph Explorer

Microsoft Graph / v1.0 reference / Security / Alerts / List alerts

Bookmark Feedback Edit Share

Version Microsoft Graph REST API v1.0

Filter by title

- > Reports
- ✓ Security
- Overview
- ✓ Alerts
 - Alerts
 - List alerts
 - Get alert
 - Update alert
- > Information protection
- > Secure scores
- > Secure score control profiles
- Errors
- > Sites and lists
- > Tasks and plans
- > Teamwork
- > Workbooks and charts

List alerts

06/04/2020 • 3 minutes to read • 5 people +3

Namespace: microsoft.graph

Retrieve a list of [alert](#) objects.

Permissions

One of the following permissions is required to call this API. To learn more, including how to choose permissions, see [Permissions](#).

Permission type	Permissions (from least to most privileged)
Delegated (work or school account)	SecurityEvents.Read.All, SecurityEvents.ReadWrite.All
Delegated (personal Microsoft account)	Not supported.
Application	SecurityEvents.Read.All, SecurityEvents.ReadWrite.All

HTTP request

HTTP

Copy

Is this page helpful?

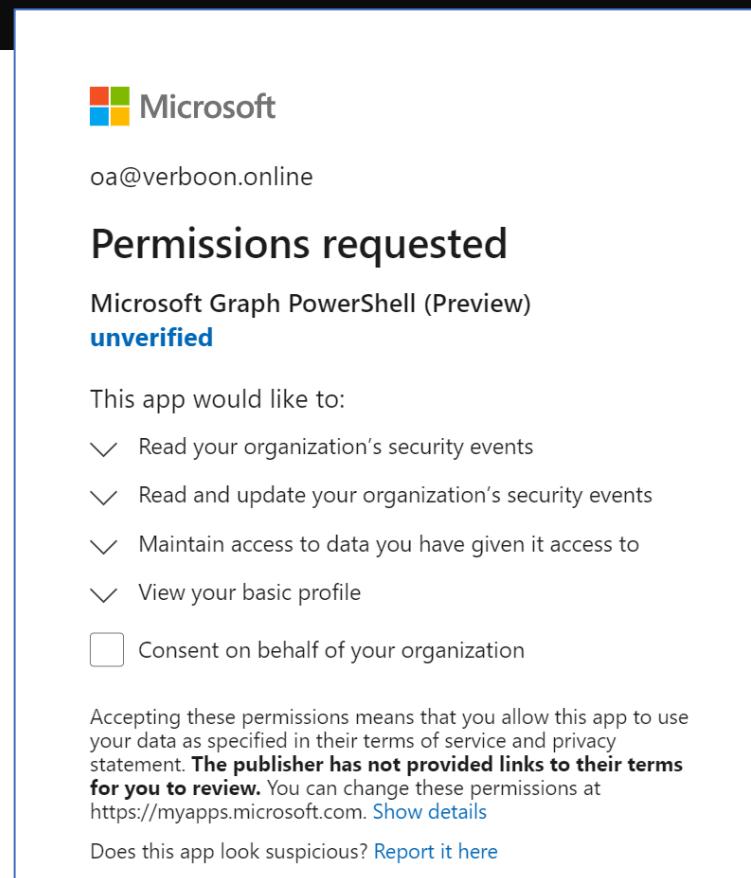
Yes No

In this article

- Permissions
- HTTP request
- Optional query parameters
- Request headers
- Request body
- Response
- Example

Managing the Microsoft Security stack with PowerShell – Microsoft Graph

```
Windows PowerShell  PowerShell Core
PS C:\> Get-MgSecurityAlert
Get-MgSecurityAlert_List: Auth token does not contain valid permissions or user does not have valid roles.
PS C:\> connect-graph -Scopes "SecurityEvents.Read.All", "SecurityEvents.ReadWrite.All" -TenantId 3-2e24ac9a9743
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code DGLXQBVNN to authenticate.
```



Managing the Microsoft Security stack with PowerShell – Microsoft Graph

```
Windows PowerShell  x  PowerShell Core  x  +  -  ×
PS C:\> Get-MgSecurityAlert | Select-object EventDateTime, Category, Title

EventDateTime          Category                                Title
-----              -----
8/18/2020 5:28:23 PM c46c4e10-5d81-409a-9fbb-3857dc9db495_d8270f6f-9dc0-4e28-a035-c65bd31b6e76 SharePointFileOperatio...
9/8/2020 6:14:07 AM Malware                                'PowerView' hacktool w...
9/8/2020 9:21:32 AM UnwantedSoftware                         'EICAR_Test_File' unwa...
9/8/2020 9:21:32 AM UnwantedSoftware                         'EICAR_Test_File' unwa...
9/8/2020 9:22:19 AM Malware                                'Wacatac' malware was ...
9/8/2020 7:53:18 AM Execution                             [Test Alert] Suspiciou...
9/8/2020 6:16:48 AM Malware                                'Sehyioa' malware was ...
9/8/2020 6:16:48 AM Malware                                'Wacatac' malware was ...
9/8/2020 6:13:28 AM DefenseEvasion                      A process was injected...
9/7/2020 8:57:46 AM Execution                             [Test Alert] Suspiciou...
9/6/2020 12:26:40 PM Malware                                'Meterpreter' malware ...
9/6/2020 12:27:57 PM Malware                                'Meterpreter' hacktool...
9/6/2020 12:38:16 PM Malware                                'Ploty' malware was pr...
9/6/2020 12:38:16 PM DefenseEvasion                      A process was injected...
9/2/2020 5:25:31 PM InitialAccess                        Connection to a custom...
9/2/2020 5:25:31 PM InitialAccess                        Connection to a custom...
9/2/2020 5:27:30 PM InitialAccess                        Connection to a custom...
9/2/2020 5:28:23 PM InitialAccess                        Connection to a custom...
9/2/2020 5:24:57 PM Persistence                           local account created
9/1/2020 2:57:31 PM Execution                            firefox usage detected
8/27/2020 9:57:33 AM LateralMovement                    Suspicious remote Powe...
8/27/2020 9:57:33 AM LateralMovement                    Suspicious remote acti...
8/27/2020 9:57:37 AM Malware                             'Injector' malware was...
8/27/2020 9:57:33 AM Execution                           Suspicious PowerShell ...
```

Managing the Microsoft Security stack with PowerShell – Microsoft Graph

```
Windows PowerShell
PS C:\Users\AlexVerboon> $signin = Get-MgAuditLogSignIn|
```

```
Windows PowerShell
PS C:\Users\AlexVerboon> $signin | Select UserDisplayName, RiskState, IPAddress | where RiskState -ne 'none'

UserDisplayName RiskState IPAddress
----- ----- -----
org admin      atRisk    212.77.61.26
org admin      atRisk    104.47.146.40
org admin      atRisk    13.93.68.229

PS C:\Users\AlexVerboon> |
```

Managing the Microsoft Security stack with PowerShell – Microsoft Graph

```
Windows PowerShell
PS C:\Users\AlexVerboon> $diraudit = Get-MgAuditLogDirectoryAudit
```

```
Windows PowerShell
PS C:\Users\AlexVerboon> $diraudit | Select ActivityDateTime, ActivityDisplayName

ActivityDateTime      ActivityDisplayName
-----              -----
9/14/2020 5:43:09 AM Consent to application
9/14/2020 5:43:08 AM Remove delegated permission grant
9/14/2020 5:43:08 AM Add delegated permission grant
9/11/2020 6:55:50 PM Update service principal
9/11/2020 6:55:50 PM Update service principal
9/8/2020 6:10:30 AM Update device
9/7/2020 9:58:49 AM Add member to group
9/7/2020 9:58:49 AM Add member to group
9/7/2020 9:58:48 AM Add member to group
9/7/2020 9:58:48 AM Add member to group
9/7/2020 9:58:47 AM Add group
9/7/2020 9:14:50 AM Update device
```

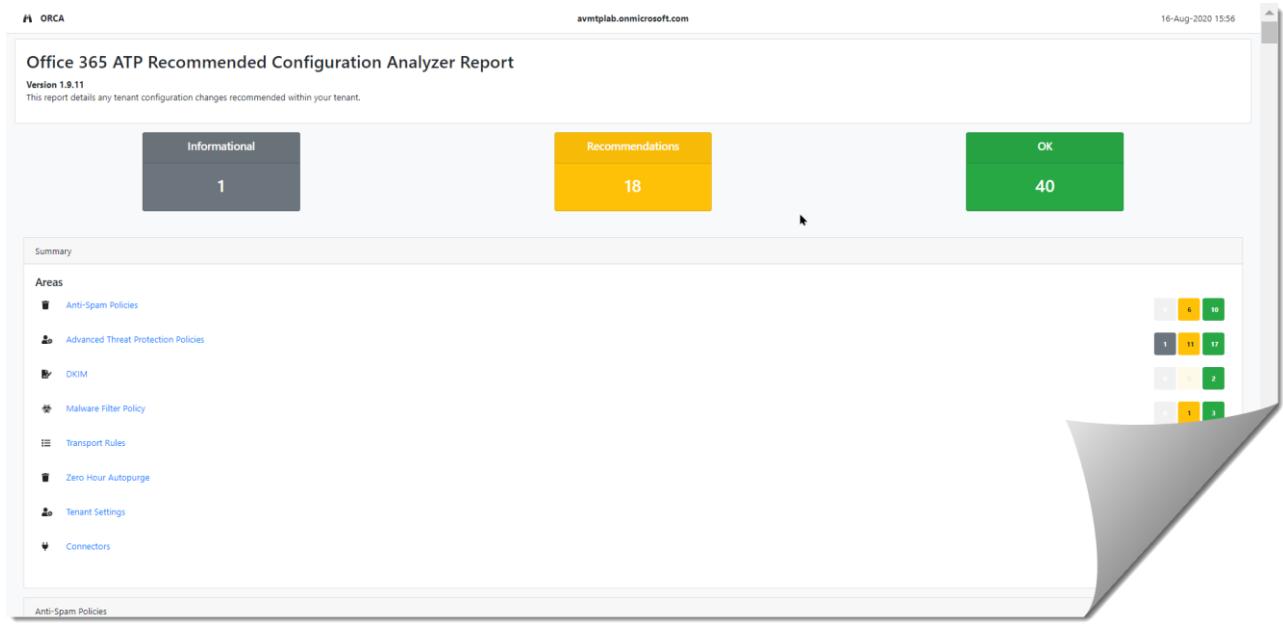
Managing the Microsoft Security stack with PowerShell – Microsoft Graph

```
Windows PowerShell
```

```
PS C:\Users\AlexVerboon> $x = Get-MgRiskyUser
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code DMBF9Y52H to authenticate.
PS C:\Users\AlexVerboon> $x
Id          IsDeleted IsProcessing RiskDetail           RiskLastUpdatedDateTime RiskL
--          -----      -----       -----           -----                -----      evel
--          -----      -----       -----           -----                -----      -----
78750083-e7a2-4f20-b35a-6f736981e860 False    False      adminConfirmedUserCompromised 7/25/2020 3:17:14 PM   high
271a9c7b-d826-4027-a907-2b0edaf256d4 False    False      none                  7/4/2020 6:06:44 PM   none
c6f5185a-c038-4bff-95be-e12642e14384 False    False      none                  9/9/2020 7:37:20 PM   low

PS C:\Users\AlexVerboon> $x | FL

History          :
Id              : 78750083-e7a2-4f20-b35a-6f736981e860
IsDeleted        : False
IsProcessing     : False
RiskDetail       : adminConfirmedUserCompromised
RiskLastUpdatedDateTime : 7/25/2020 3:17:14 PM
RiskLevel        : high
RiskState         : confirmedCompromised
UserDisplayName   : Adele Vance
UserPrincipalName : AdeleV@avmtplab.onmicrosoft.com
Keys             : {}
Values            : {}
AdditionalProperties : {}
Count            : 0
```



**The Office 365 ATP Recommended Configuration Analyzer
(ORCA)**
<https://github.com/cammurray/orca>

Managing the Microsoft Security stack with PowerShell – Office 365 ATP Recommended Configuration Analyzer

The diagram illustrates a sequence of PowerShell sessions:

- Session 1:** A PowerShell window titled "Windows PowerShell" shows the command `find-module -name Orca`. The output is a table:

Version	Name	Repository	Description
1.9.11	ORCA	PSGallery	Office 365 Advanced Threat Prote...

- Session 2:** A PowerShell window titled "Windows PowerShell" shows the command `find-module -name Orca | install-module -Scope CurrentUser`.
- Session 3:** A PowerShell window titled "Windows PowerShell" shows the command `Get-ORCAResult`.
- Session 4:** A PowerShell window titled "Windows PowerShell" shows the command `Get-ORCAResult` followed by a large block of log output. A blue arrow points from Session 3 to Session 4.

The log output in Session 4 includes numerous entries such as:

- 08/16/2020 15:56:04 Analysis - Anti-Spam Policies - IP Allow Lists
- 08/16/2020 15:56:04 Analysis - Anti-Spam Policies - Advanced Spam Filter (ASF)
- 08/16/2020 15:56:04 Analysis - Anti-Spam Policies - Outbound spam filter policy settings
- 08/16/2020 15:56:04 Analysis - Anti-Spam Policies - End-user Spam notifications
- 08/16/2020 15:56:04 Analysis - Anti-Spam Policies - Quarantine retention period
- 08/16/2020 15:56:04 Analysis - Anti-Spam Policies - High Confidence Phish Action
- 08/16/2020 15:56:04 Analysis - Anti-Spam Policies - Allowed Senders
- 08/16/2020 15:56:04 Analysis - Anti-Spam Policies - High Confidence Spam Action
- 08/16/2020 15:56:05 Analysis - Anti-Spam Policies - Bulk Complaint Level
- 08/16/2020 15:56:05 Analysis - Anti-Spam Policies - Spam Action
- 08/16/2020 15:56:05 Analysis - Anti-Spam Policies - Mark Bulk as Spam
- 08/16/2020 15:56:05 Analysis - Connectors - Domains
- 08/16/2020 15:56:05 Analysis - Connectors - Enhanced Filtering Configuration
- 08/16/2020 15:56:05 Analysis - DKIM - Signing Configuration
- 08/16/2020 15:56:05 Analysis - DKIM - DNS Records
- 08/16/2020 15:56:08 Analysis - Malware Filter Policy - External Sender Notifications
- 08/16/2020 15:56:08 Analysis - Malware Filter Policy - Malware Filter Policy Policy Rules
- 08/16/2020 15:56:08 Analysis - Malware Filter Policy - Common Attachment Type Filter
- 08/16/2020 15:56:09 Analysis - Malware Filter Policy - Internal Sender Notifications
- 08/16/2020 15:56:09 Analysis - Tenant Settings - Unified Audit Log
- 08/16/2020 15:56:09 Analysis - Transport Rules - Domain Whitelisting
- 08/16/2020 15:56:09 Analysis - Zero Hour Autopurge - Supported filter policy action
- 08/16/2020 15:56:09 Analysis - Zero Hour Autopurge - Zero Hour Autopurge Enabled for Malware
- 08/16/2020 15:56:09 Analysis - Zero Hour Autopurge - Zero Hour Autopurge Enabled for Phish
- 08/16/2020 15:56:09 Analysis - Zero Hour Autopurge - Zero Hour Autopurge Enabled for Spam
- 08/16/2020 15:56:09 Generating Output
- 08/16/2020 15:56:09 Output - HTML
- 08/16/2020 15:56:11 Complete! Output is in C:\Users\AlexVerboon\AppData\Local\Microsoft\ORCA\ORCA-avmtplab-202008161556.html

Managing the Microsoft Security stack with PowerShell – Office 365 ATP Recommended Configuration Analyzer

ORCA avmtplab.onmicrosoft.com 16-Aug-2020 15:56

Office 365 ATP Recommended Configuration Analyzer Report

Version 1.9.11
This report details any tenant configuration changes recommended within your tenant.

Informational

1

Recommendations

18

OK

40

Summary

Areas

Area	Informational	Recommendations	OK
Anti-Spam Policies	0	6	10
Advanced Threat Protection Policies	1	11	17
DKIM	0	0	2
Malware Filter Policy	0	1	3
Transport Rules	0	0	1
Zero Hour Autopurge	0	0	4
Tenant Settings	0	0	1



Office 365 Advanced Threat Protection

Managing the Microsoft Security stack with PowerShell – Office ATP

```
Windows PowerShell  x  PowerShell Core  x  +  -  ×
PS C:\> connect-exchangeonline

-----
The module allows access to all existing remote PowerShell (V1) cmdlets in addition to the 9 new, faster, and more reliable cmdlets.

-----

| Old Cmdlets                 | New/Reliable/Faster Cmdlets    |
|-----------------------------|--------------------------------|
| Get-CASMailbox              | Get-EXOCASMailbox              |
| Get-Mailbox                 | Get-EXOMailbox                 |
| Get-MailboxFolderPermission | Get-EXOMailboxFolderPermission |
| Get-MailboxFolderStatistics | Get-EXOMailboxFolderStatistics |
| Get-MailboxPermission       | Get-EXOMailboxPermission       |
| Get-MailboxStatistics       | Get-EXOMailboxStatistics       |
| Get-MobileDeviceStatistics  | Get-EXOMobileDeviceStatistics  |
| Get-Recipient               | Get-EXORecipient               |
| Get-RecipientPermission     | Get-EXORecipientPermission     |

  
-----  
To get additional information, run: Get-Help Connect-ExchangeOnline or check https://aka.ms/exops-docs  
Send your product improvement suggestions and feedback to exocmdletpreview@service.microsoft.com. For issues related to the module, contact Microsoft support. Don't use the feedback alias for problems or support issues.  
-----  
PS C:\> |
```

Managing the Microsoft Security stack with PowerShell – Office ATP - Safe Attachments

The screenshot displays two windows side-by-side. On the left is a Windows PowerShell window showing the command `Get-SafeAttachmentPolicy | where Name -eq "Global - Safe Attachments" | fl`. The output lists various policy settings, with several fields highlighted by yellow boxes: `RedirectAddress`, `Action`, and `Name`. Arrows from these highlighted fields point to the corresponding configuration options in the adjacent dialog box. The right window is titled "Global - Safe Attachments" and "Editing Settings". It contains descriptive text about monitoring, replacing, and blocking attachments, and a list of five action options: Off, Monitor, Block, Replace, and Dynamic Delivery (which is selected). Below this is a "Redirect attachment on detection" section with a checked checkbox for "Enable redirect" and a text input field containing the email address `mvpadmin@avmtplab.onmicrosoft.com`.

```
PS C:\> Get-SafeAttachmentPolicy | where Name -eq "Global - Safe Attachments" | fl
```

Property	Value
RunspaceId	: 54f67ea9-29e3-42e7-a599-a6583c4f96f5
RedirectAddress	: mvpadmin@avmtplab.onmicrosoft.com
Redirect	: True
Action	: DynamicDelivery
ScanTimeout	: 30
ConfidenceLevelThreshold	: 80
OperationMode	: Delay
Enable	: True
ActionOnError	: True
RecommendedPolicyType	: Custom
IsDefault	: False
AdminDisplayName	: Safe Attachments policy for all users
EnableOrganizationBranding	: False
ExchangeVersion	: 0.20 (15.0.0.0)
Name	: Global - Safe Attachments
DistinguishedName	: CN=Global - Safe Attachments,CN=Safe Aoft.com,CN=ConfigurationUnits,DC=CHEP2
Identity	: Global - Safe Attachments
ObjectCategory	: CHEP278A003.PROD.OUTLOOK.COM/Configura
ObjectClass	: {top, msExchSafeAttachmentProtectionCo
WhenChanged	: 6/29/2020 8:43:57 AM
WhenCreated	: 6/29/2020 8:43:45 AM
WhenChangedUTC	: 6/29/2020 6:43:57 AM
WhenCreatedUTC	: 6/29/2020 6:43:45 AM
ExchangeObjectId	: 35477c0e-b2ab-411d-b5f4-8352a2af18f5
OrganizationId	: CHEP278A003.PROD.OUTLOOK.COM/Microsoft CHEP278A003.PROD.OUTLOOK.COM/Configura
Id	: Global - Safe Attachments
Guid	: 35477c0e-b2ab-411d-b5f4-8352a2af18f5
OriginatingServer	: DB6P278A03DC002.CHEP278A003.PROD.OUTLO
IsValid	: True
ObjectState	: Unchanged

Global - Safe Attachments
Editing Settings

Monitor, Replace and Block actions may cause significant delay to email delivery. [Learn more](#)
Dynamic Delivery is only available for recipients with hosted mailboxes. [Learn more](#)
If you choose the Block, Replace or Dynamic Delivery options and malware is detected in attachment, the message containing the attachment will be quarantined and can be released only by an admin.

- Off - Attachment will not be scanned for malware.
- Monitor - Continue delivering the message after malware is detected; track scan results.
- Block - Block the current and future email and attachments with detected malware.
- Replace - Block the attachments with detected malware, continue to deliver the message.
- Dynamic Delivery (Preview Feature)- Deliver the message without attachments immediately and reattach once scan is complete.

Redirect attachment on detection

Send the blocked, monitored, or replaced attachment to an email address.

Enable redirect (i)

Send the attachment to the following email address *

Managing the Microsoft Security stack with PowerShell – Office ATP – Safe Attachments

The screenshot displays two windows side-by-side. On the left is a Windows PowerShell window showing PowerShell commands and their output. On the right is a Microsoft 365 Admin Center dialog titled "Global - Safe Attachments" with the sub-section "Editing Applied to".

PowerShell Window Output:

```
PS C:\> $rule = Get-SafeAttachmentPolicy | where Name -eq "Global - Safe Attachments" | Get-SafeAttachmentRule
PS C:\> $rule | fl
```

	:	
RunspaceId	:	084db004-fe56-4b8f-9698-067a1af0eb57
SafeAttachmentPolicy	:	Global - Safe Attachments
State	:	Enabled
Priority	:	0
Comments	:	
Description	:	If the message: recipients's address domain portion below 'avmtplab.onmicrosoft.com' Take the following actions: Apply safe attachment policy "Global - S
RuleVersion	:	15.0.5.2
SentTo	:	
SentToMemberOf	:	
RecipientDomainIs	:	{verboon.online, avmtplab.onmicrosoft.com}
ExceptIfSentTo	:	
ExceptIfSentToMemberOf	:	
ExceptIfRecipientDomainIs	:	
Conditions	:	{Microsoft.Exchange.MessagingPolicies.Rules.}
Exceptions	:	
Identity	:	Global - Safe Attachments
DistinguishedName	:	CN=Global - Safe Attachments,CN=SafeAttachmentPolicy,CN=ConfigurationUnit,c0f4dadb-9dfe-40fc-a897-ed5ccd69ceea
Guid	:	c0f4dadb-9dfe-40fc-a897-ed5ccd69ceea
ImmutableId	:	CHEP278A003.PROD.OUTLOOK.COM/Microsoft Exchange
OrganizationId	:	CHEP278A003.PROD.OUTLOOK.COM/ConfigurationUnit
Name	:	Global - Safe Attachments
IsValid	:	True
WhenChanged	:	6/29/2020 8:43:57 AM
ExchangeVersion	:	0.1 (8.0.535.0)
ObjectState	:	Unchanged

Microsoft 365 Admin Center Dialog:

Global - Safe Attachments
Editing Applied to

Specify the users, groups, or domains for whom this policy applies by creating recipient based rules:

Any of these
verboon.online,avmtplab.onmicrosoft.com
Choose
(2 selected)

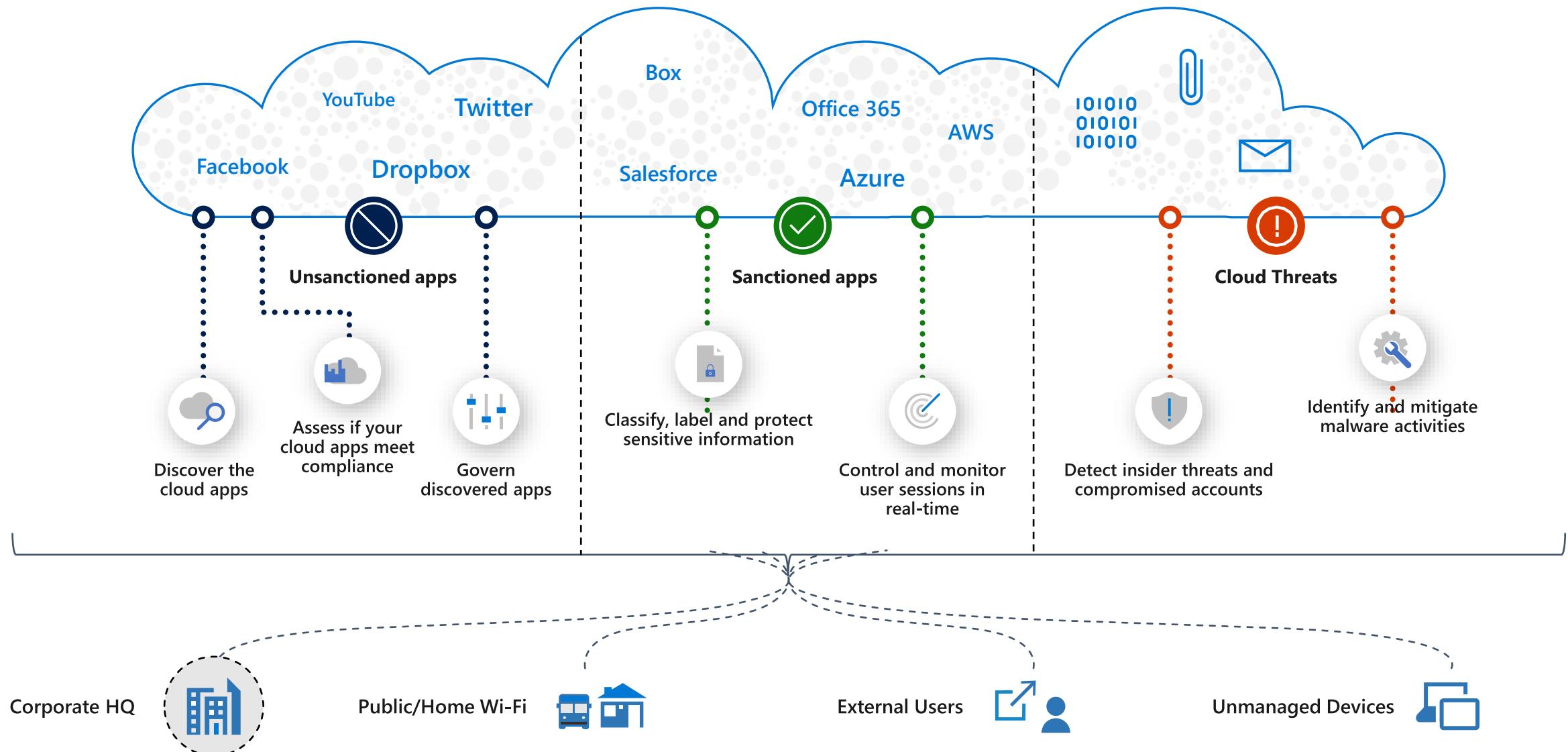
+ Add a condition ▾

Save Cancel



Unofficial Microsoft Cloud App Security PowerShell Module
<https://github.com/Microsoft/MCAS>

Managing the Microsoft Security stack with PowerShell – Microsoft Cloud App Security



Managing the Microsoft Security stack with PowerShell – Microsoft Cloud App Security

The image shows two separate Windows PowerShell windows. The left window displays the command `PS C:\> find-module -name mcas | install-module -Scope CurrentUser`. The right window displays the command `PS C:\> find-command -ModuleName MCAS`, which lists all the cmdlets available in the MCAS module, each with its name, version (3.3.6), module name (MCAS), and repository (PSGallery).

Name	Version	ModuleName	Repository
Add-MCASAdminAccess	3.3.6	MCAS	PSGallery
ConvertFrom-MCASTimestamp	3.3.6	MCAS	PSGallery
Export-MCASBlockScript	3.3.6	MCAS	PSGallery
Export-MCASCredential	3.3.6	MCAS	PSGallery
Get-MCASAccount	3.3.6	MCAS	PSGallery
Get-MCASActivity	3.3.6	MCAS	PSGallery
Get-MCASActivityType	3.3.6	MCAS	PSGallery
Get-MCASAdminAccess	3.3.6	MCAS	PSGallery
Get-MCASAlert	3.3.6	MCAS	PSGallery
Get-MCASAppId	3.3.6	MCAS	PSGallery
Get-MCASAppInfo	3.3.6	MCAS	PSGallery
Get-MCASAppPermission	3.3.6	MCAS	PSGallery
Get-MCASConfiguration	3.3.6	MCAS	PSGallery
Get-MCASCredential	3.3.6	MCAS	PSGallery
Get-MCASDiscoveredApp	3.3.6	MCAS	PSGallery
Get-MCASDiscoveryDataSource	3.3.6	MCAS	PSGallery
Get-MCASDiscoverySampleLog	3.3.6	MCAS	PSGallery
Get-MCASFile	3.3.6	MCAS	PSGallery
Get-MCASGovernanceAction	3.3.6	MCAS	PSGallery
Get-MCASPTag	3.3.6	MCAS	PSGallery
Get-MCASLogCollector	3.3.6	MCAS	PSGallery
Get-MCASPolicy	3.3.6	MCAS	PSGallery
Get-MCASPortalSettings	3.3.6	MCAS	PSGallery
Get-MCASSiemAgent	3.3.6	MCAS	PSGallery
Get-MCASStream	3.3.6	MCAS	PSGallery
Get-MCASSubnetCollection	3.3.6	MCAS	PSGallery
Get-MCASUserGroup	3.3.6	MCAS	PSGallery
Import-MCASCredential	3.3.6	MCAS	PSGallery
Install-MCASSiemAgent	3.3.6	MCAS	PSGallery
New-MCASDiscoveryDataSource	3.3.6	MCAS	PSGallery
New-MCASSiemAgentToken	3.3.6	MCAS	PSGallery
New-MCASSubnetCollection	3.3.6	MCAS	PSGallery
Remove-MCASAdminAccess	3.3.6	MCAS	PSGallery
Remove-MCASDiscoveryDataSource	3.3.6	MCAS	PSGallery
Remove-MCASSubnetCollection	3.3.6	MCAS	PSGallery
Send-MCASDiscoveryLog	3.3.6	MCAS	PSGallery
Set-MCASAlert	3.3.6	MCAS	PSGallery

Managing the Microsoft Security stack with PowerShell – Microsoft Cloud App Security

The screenshot shows the Microsoft Cloud App Security interface. The top navigation bar includes 'Cloud App Security' and a search bar. On the left, there's a sidebar with icons for Home, Security extensions, SIEM agents, External DLP, Playbooks, and API tokens (which is highlighted with a yellow box). The main content area is titled 'Generate new token'. It has a 'Token name:' field containing 'MCASPowerShell' (also highlighted with a yellow box). A success message box is overlaid on the page, stating 'API token was successfully generated' and providing a URL: 'https://m365x600058.eu2.portal.cloudappsecurity.com'. The URL field is also highlighted with a yellow box. The top right corner of the main window has a gear icon, a help icon, and a settings icon.

Cloud App Security

Security extensions

API tokens

SIEM agents

External DLP

Playbooks

Token name

Generate new token

Token name: *

MCASPowerShell

Learn more

Generate new token

API token was successfully generated

Your URL is: https://m365x600058.eu2.portal.cloudappsecurity.com

Close

SYSTEM

Settings

Governance log

Security extensions

Manage admin access

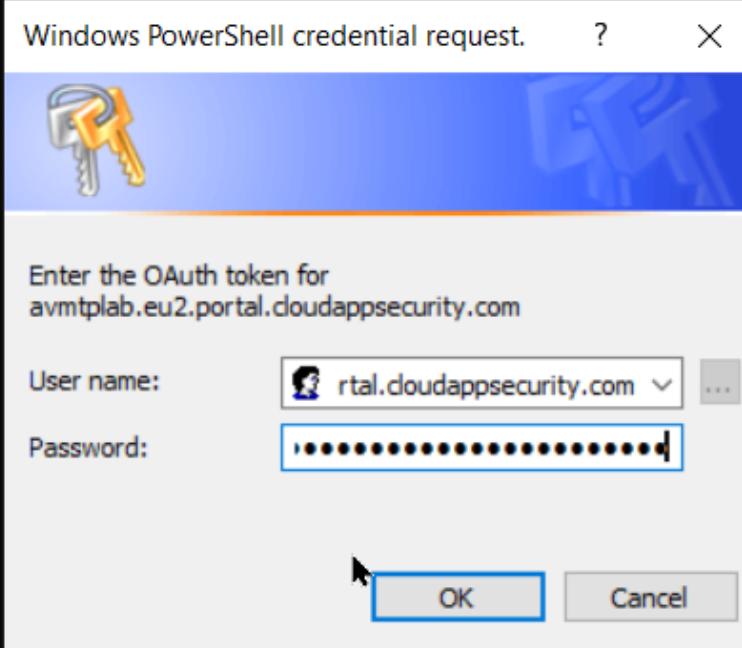
Managing the Microsoft Security stack with PowerShell – Microsoft Cloud App Security

```
Windows PowerShell x + v

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\AlexVerboon>
PS C:\Users\AlexVerboon> Get-MCASCredential -TenantUri "avmtplab.eu2.portal.cloudappsecurity.com"

Windows PowerShell credential request. ? X


Enter the OAuth token for  
avmtplab.eu2.portal.cloudappsecurity.com



User name:  ...



Password:



OK Cancel


```

Managing the Microsoft Security stack with PowerShell – Microsoft Cloud App Security

```
Windows PowerShell  x + v - □ ×

PS C:\Users\AlexVerboon> Get-MCASAlert -Severity High
PS C:\Users\AlexVerboon> Get-MCASAlert

_id          : 5f59331cba4289d060343a12
timestamp    : 1599680866616
entities     : {@{policyType=ANOMALY_DETECTION; id=5ef5fa700e5b1799cb74f26d; label=Activity from infrequent country; type=policyRule}, @{id=11161; type=service; label=Office 365}, @{p=oa@verboon.online; saas=11161; entityType=2; inst=0; label=org admin; id=c6f5185a-c038-4bff-95be-e12642e14384; type=account}, @{label=LI; id=LI; type=country}...}
title        : Activity from infrequent country
description  : <p>org admin (oa@verboon.online) performed an activity. No activity was performed in Liechtenstein in the past 75 days.</p>
stories      : {0}
contextId   : f50b6fb5-d3bc-4cff-b0b3-2e24ac9a9743
threatScore  : 30
isSystemAlert: False
isPreview    : False
severityValue: 1
statusValue  : 0
idValue      : 15859713
URL          : https://avmtplab.portal.cloudappsecurity.com/#/alerts/5f59331cba4289d060343a12
Identity    : 5f59331cba4289d060343a12

_id          : 5f157bc4289d0606e8897
timestamp    : 1595243044927
entities     : {@{label=adelev@avmtplab.onmicrosoft.com; id=adelev@avmtplab.onmicrosoft.com; type=user}, @{policyType=ANOMALY_DETECTION; id=5ef5fa700e5b1799cb74f242; label=Activity from anonymous IP addresses; type=policyRule}, @{id=18.27.197.252; type=ip; triggeredAlert=True; label=18.27.197.252}, @{id=11161; type=service; label=Office 365}...}
title        : Activity from a Tor IP address
description  : <p>A failed sign in was detected from a Tor IP address<br>The Tor IP address 18.27.197.252 was used by Adele Vance (adelev@avmtplab.onmicrosoft.com).</p>
stories      : {0}
contextId   : f50b6fb5-d3bc-4cff-b0b3-2e24ac9a9743
```

Managing the Microsoft Security stack with PowerShell – Microsoft Cloud App Security

```
Windows PowerShell x + v
PS C:\Users\AlexVerboon> Get-MCASPolicy | Select Name,Description | Format-Table -AutoSize -Wrap

name                                     description
----                                     -----
Suspicious OAuth app file download activities
Preview: Suspicious change of CloudTrail logging service
Suspicious email deletion activity (by user)
Preview: Suspicious Power BI report sharing
Multiple VM creation activities
Preview: Multiple Power BI report sharing activities
Multiple storage deletion activities
```

Managing the Microsoft Security stack with PowerShell – Microsoft Cloud App Security

```
Windows PowerShell

PS C:\Users\AlexVerboon> Get-MCASActivity -EventTypeName "EVENT_CATEGORY_ACCESS_FILE"

_id : 98518865_20892_2fe5f06c-ca1f-4421-89dc-08d84f071100
tenantId : 98518865
aadTenantId : f50b6fb5-d3bc-4cff-b0b3-2e24ac9a9743
appId : 20892
saasId : 20892
timestamp : 1599027104000
timestampRaw : 1599027104000
instantiation : 1599031498302
instantiationRaw : 1599031498302
created : 1599032141289
createdRaw : 1599032141289
eventType : 200827
eventTypeValue : EVENT_0365_SP_FILE_ACCESSED
eventRouting : @{scubaUnpacker=False; lograber=True; auditing=True}
device : @{clientIP=52.104.8.157; userAgent=OfficeWordWRS; countryCode=CH}
location : @{countryCode=CH; city=Zurich; postalCode=8052; regionCode=ZH; region=Zurich; longitude=8.5546; latitude=47.3664; organizationSearchable=SharePoint Online and OneDrive for Business server; anonymousProxy=False; isSatelliteProvider=False; ipTags=System.Object[]; category=5; categoryValue=CLOUD_PROXY_NETWORK_IP}
user : @{userName=oa@verboon.online; userTags=System.Object[]}
userAgent : @{family=UNKNOWN; name=Unknown; operatingSystem=; type=Unknown; typeName=Unknown; deviceType=OTHER; nativeBrowser=False; os=OTHER; browser=UNKNOWN}
internals : @{otherIPs=System.Object[]}
mainInfo : @{eventObjects=System.Object[]; rawOperationName=FileAccessed; prettyOperationName=FileAccessed; type=view}
confidenceLevel : 30
source : 2
lograberService : @{o365EventGrabber=True; gediEvent=True}
srcAppId : 11161
collected : @{o365=}
rawDataJson : @{OrganizationId=f50b6fb5-d3bc-4cff-b0b3-2e24ac9a9743; SourceFileExtension=docx; HighPriorityMediaProcessing=False}
```

Managing the Microsoft Security stack with PowerShell – **Microsoft Cloud App Security**

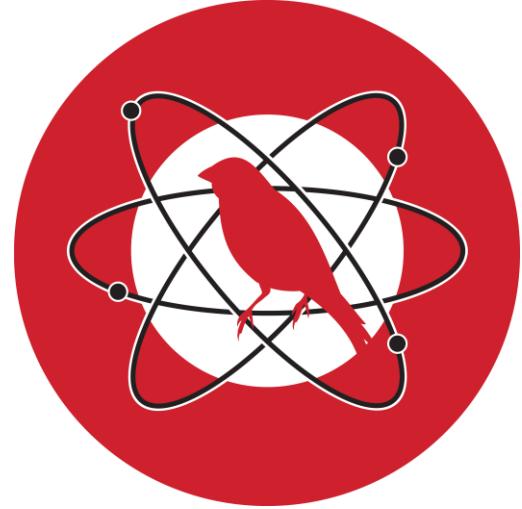
AzureAD Conditional Access

Managing the Microsoft Security stack with PowerShell – [AzureAD Conditional Access](#)

Document Conditional Access with PowerShell @nicolonsky, Nicolas Suter

<https://github.com/nicolonsky/ConditionalAccessDocumentation>

```
PS C:\Dev\Alex\MSSecurityPowerShell\AzureAD\ConditionalAccess> connect-graph
Welcome To Microsoft Graph!
PS C:\Dev\Alex\MSSecurityPowerShell\AzureAD\ConditionalAccess> .\Invoke-ConditionalAccessDocumentation.ps1
Exported Documentation to 'C:\Dev\Alex\MSSecurityPowerShell\AzureAD\ConditionalAccess\ConditionalAccessDocumentation.csv'
PS C:\Dev\Alex\MSSecurityPowerShell\AzureAD\ConditionalAccess> |
```

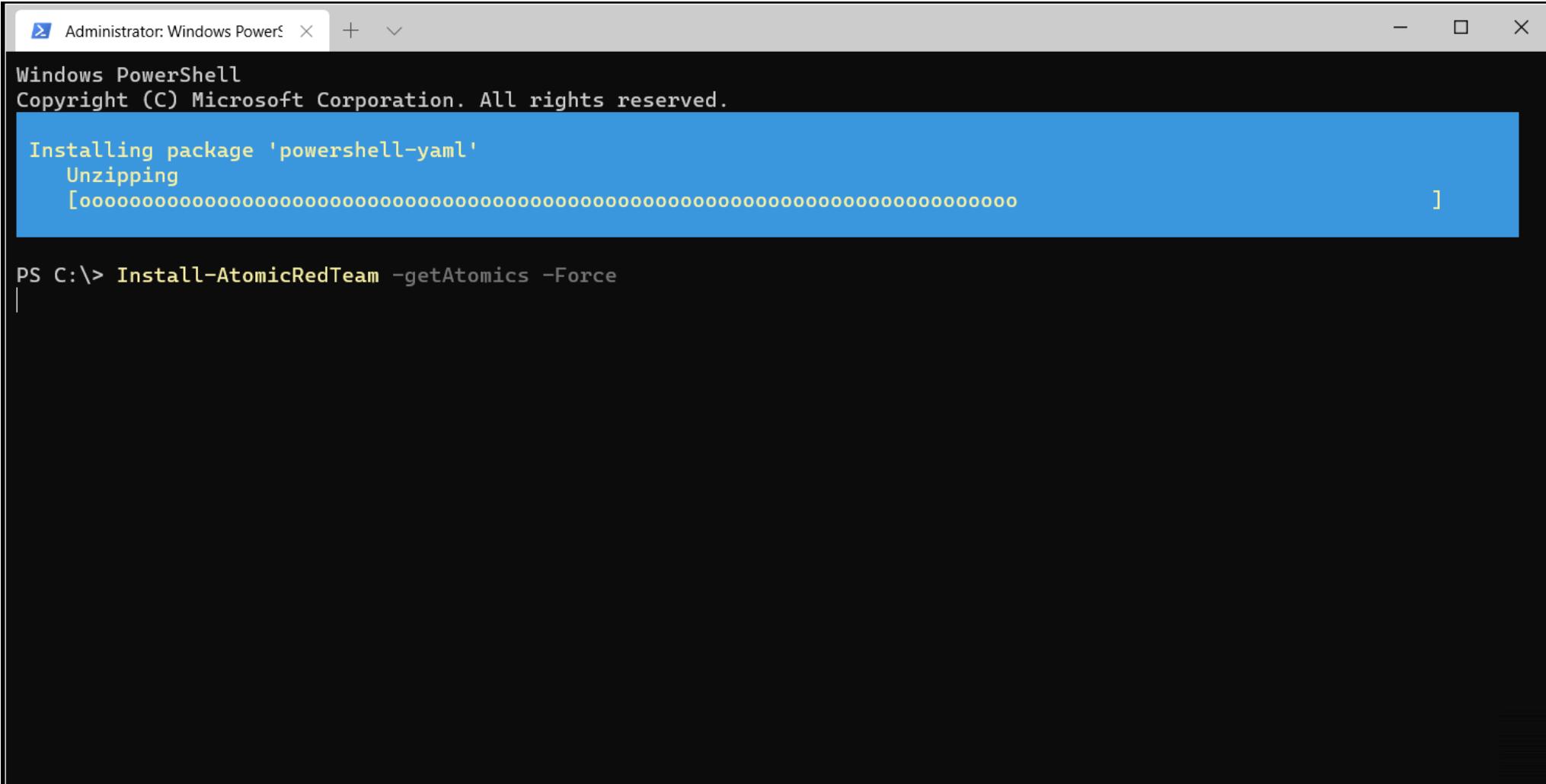


Attack Simulation with PowerShell

<https://github.com/redcanaryco/Invoke-AtomicRedTeam/wiki>

Managing the Microsoft Security stack with PowerShell – **Atomic execution framework**

```
IEX (IWR 'https://raw.githubusercontent.com/redcanaryco/invoke-atomicredteam/master/install-atomicredteam.ps1');  
Install-AtomicRedTeam -getAtomics -Force
```



The screenshot shows an Administrator Windows PowerShell window. The title bar reads "Administrator: Windows PowerShell". The window displays the following text:

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Installing package 'powershell-yaml'  
Unzipping  
[oooooooooooooooooooooooooooooooooooooooooooo]
```

At the bottom of the window, the command "PS C:\> Install-AtomicRedTeam -getAtomics -Force" is visible, indicating the continuation of the process.

Managing the Microsoft Security stack with PowerShell – Atomic execution framework

```
Invoke-AtomicTest All -ShowDetailsBrief
```

```
Administrator: Windows PowerShell + × - ×
T1562.001-19 Stop and Remove Arbitrary Security Windows Service
T1562.001-20 Uninstall Crowdstrike Falcon on Windows
T1562.001-21 Tamper with Windows Defender Evade Scanning -Folder
T1562.001-22 Tamper with Windows Defender Evade Scanning -Extension
T1562.001-23 Tamper with Windows Defender Evade Scanning -Process
T1562.002-1 Disable Windows IIS HTTP Logging
T1562.004-2 Disable Microsoft Defender Firewall
T1562.004-3 Allow SMB and RDP on Microsoft Defender Firewall
T1562.004-4 Opening ports for proxy - HARDRAIN
T1564.001-3 Create Windows System File with Attrib
T1564.001-4 Create Windows Hidden File with Attrib
T1564.003-1 Hidden Window
T1564.004-1 Alternate Data Streams (ADS)
T1564.004-2 Store file in Alternate Data Stream (ADS)
T1564.004-3 Create ADS command prompt
T1564.004-4 Create ADS PowerShell
T1566.001-1 Download Phishing Attachment - VBScript
T1566.001-2 Word spawned a command shell and used an IP address in the command line
T1569.002-1 Execute a Command as a Service
T1569.002-2 Use PsExec to execute a command on a remote host
T1571-1 Testing usage of uncommonly used port with PowerShell
T1573-1 OpenSSL C2
T1574.001-1 DLL Search Order Hijacking - amsi.dll
T1574.002-1 DLL Side-Loading using the Notepad++ GUP.exe binary
T1574.009-1 Execution of program.exe as service with unquoted service path
T1574.010-1 File System Permissions Weakness
T1574.011-1 Service Registry Permissions Weakness
PS C:\>
```

Managing the Microsoft Security stack with PowerShell – Atomic execution framework

```
Administrator: Windows PowerShell + ×  
PS C:\> Invoke-AtomicTest All -ShowDetailsBrief  
PathToAtomicsFolder = C:\AtomicRedTeam\atomics  
  
T1003-1 Powershell Mimikatz  
T1003-2 Gsecdump  
T1003-3 Credential Dumping with NPPSpy  
T1003.001-1 Windows Credential Editor  
T1003.001-2 Dump LSASS.exe Memory using ProcDump  
T1003.001-3 Dump LSASS.exe Memory using comsvcs.dll  
T1003.001-4 Dump LSASS.exe Memory using direct system calls and API unhooking  
T1003.001-6 Offline Credential Theft With Mimikatz  
T1003.001-7 LSASS read with pypykatz  
T1003.002-1 Registry dump of SAM, creds, and secrets  
T1003.002-2 Registry parse with pypykatz  
T1003.002-3 esentutl.exe SAM copy  
T1003.003-1 Create Volume Shadow Copy  
T1003.003-2 Copy NTDS.dit from Volume  
T1003.003-3 Dump Active Directory Data  
T1003.003-4 Create Volume Shadow Copy  
T1003.003-5 Create Volume Shadow Copy  
T1003.003-6 Create Symlink to Volume S  
T1003.004-1 Dumping LSA Secrets  
T1007-1 System Service Discovery  
T1007-2 System Service Discovery - net
```

```
Administrator: Windows PowerShell + × — □ ×  
PS C:\> Invoke-AtomicTest -AtomicTechnique T1003.002 -TestNumbers 1 -ShowDetails  
PathToAtomicsFolder = C:\AtomicRedTeam\atomics  
  
[*****BEGIN TEST*****]  
Technique: OS Credential Dumping: Security Account Manager T1003.002  
Atomic Test Name: Registry dump of SAM, creds, and secrets  
Atomic Test Number: 1  
Atomic Test GUID: 5c2571d0-1572-416d-9676-812e64ca9f44  
Description: Local SAM (SAM & System), cached credentials (System & Security) and LSA secrets (System & Security) can be enumerated via three registry keys. Then processed locally using https://github.com/Neohapsis/creddump7  
Upon successful execution of this test, you will find three files named, sam, system and security in the %temp% directory.  
  
Attack Commands:  
Executor: command_prompt  
ElevationRequired: True  
Command:  
reg save HKLM\sam %temp%\sam  
reg save HKLM\system %temp%\system  
reg save HKLM\security %temp%\security  
  
Cleanup Commands:  
Command:  
del %temp%\sam >nul 2> nul  
del %temp%\system >nul 2> nul  
del %temp%\security >nul 2> nul  
[!!!!!!END TEST!!!!!!]  
  
PS C:\> |
```

Managing the Microsoft Security stack with PowerShell – Atomic execution framework

Execute the simulation

The screenshot illustrates the execution of an atomic test via PowerShell and its detection by Microsoft Defender ATP.

PowerShell Session:

```
PS C:\> Invoke-AtomicTest -AtomicTechnique T1003.002 -TestNumbers 1
PathToAtomsicsFolder = C:\AtomicRedTeam\atomics

Executing test: T1003.002-1 Registry dump of SAM, creds, and secrets

Done executing test: T1003.002-1
PS C:\>
```

Microsoft Defender Security Center Alert:

The alert details the following information:

- Device:** Iclient02 (Risk level: High)
- User:** Iclient02\administrator
- Data sensitivity:** Top Secret
- PE metadata:** reg.exe
- Referenced in co...:** security
- Alert Story:** Suspicious sequence of exploration activities, Sensitive information lookup
- Details:** Suspicious registry export (Medium, New)
- Actions:** Open alert page, See in timeline, Link to another incident, D. Contact your incident response team, or contact Microsoft support for investigation and remediation services.

Bottom Panel:

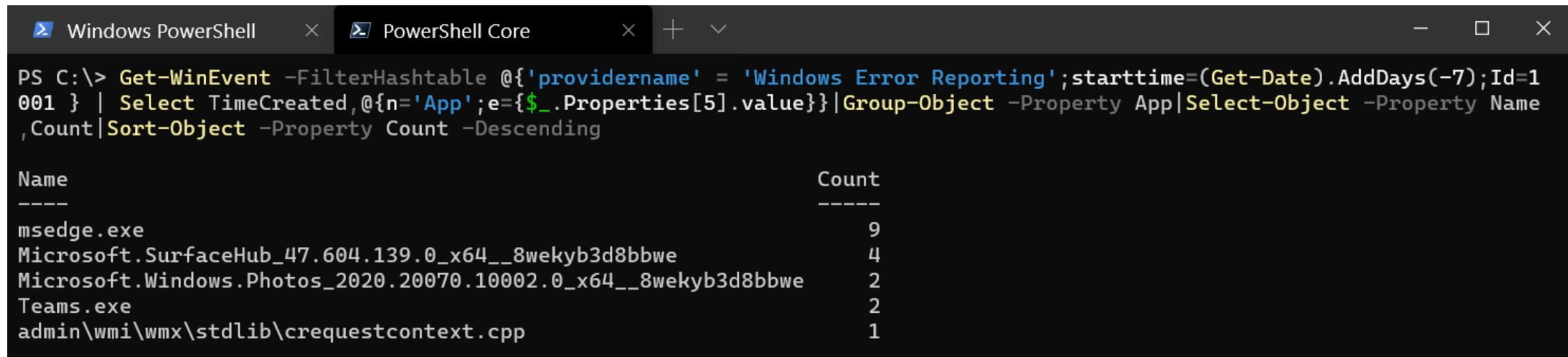
A summary of the detected activity:

- Suspicious registry export**
- Process:** [5872] cmd.exe /c "reg save HKLM\sam %temp%\sam & reg save HKLM\system %temp%\system & reg..."
- Image file path:** C:\Windows\System32\reg.exe
- Image file SHA1:** c0db341defa8ef40c03ed769a5001d600e0f4dae
- Comments and history:** (empty)

Random snippets I found on the internet or created myself

Managing the Microsoft Security stack with PowerShell – Application Crashes

```
Get-WinEvent -FilterHashtable @{'providername' = 'Windows Error Reporting'; starttime=(Get-Date).AddDays(-7); Id=1001 } | Select TimeCreated,@{n='App';e={$_.Properties[5].value}}|Group-Object -Property App|Select-Object -Property Name,Count|Sort-Object -Property Count -Descending
```



The screenshot shows a Windows PowerShell window with two tabs: "Windows PowerShell" and "PowerShell Core". The "Windows PowerShell" tab is active and displays a command and its output. The command is:

```
PS C:\> Get-WinEvent -FilterHashtable @{'providername' = 'Windows Error Reporting'; starttime=(Get-Date).AddDays(-7); Id=1001 } | Select TimeCreated,@{n='App';e={$_.Properties[5].value}}|Group-Object -Property App|Select-Object -Property Name,Count|Sort-Object -Property Count -Descending
```

The output is a table showing the count of application crashes:

Name	Count
---	----
msedge.exe	9
Microsoft.SurfaceHub_47.604.139.0_x64__8wekyb3d8bbwe	4
Microsoft.Windows.Photos_2020.20070.10002.0_x64__8wekyb3d8bbwe	2
Teams.exe	2
admin\wmi\wmx\stdlib\crequestcontext.cpp	1

Managing the Microsoft Security stack with PowerShell – Importing Microsoft Security Baselines

<https://www.verboon.info/2019/10/importing-gpo-security-baselines-with-powershell/>

The screenshot displays two windows side-by-side. On the left is a Windows PowerShell window titled 'Administrator: Windows PowerShell' with the command PS C:\temp> Import-SecurityBaselineGPO -GPOBackupPath C:\temp\secbaselines\Windows10_1903\GPOs -verbose. The output shows verbose processing of various GPO objects across different targets (e.g., Credential Guard, Internet Explorer 11, Computer, User, Domain Security, Domain Controller Virtualization Based Security) on MSFT Windows 10 1903 and Server 1903 Member Servers.

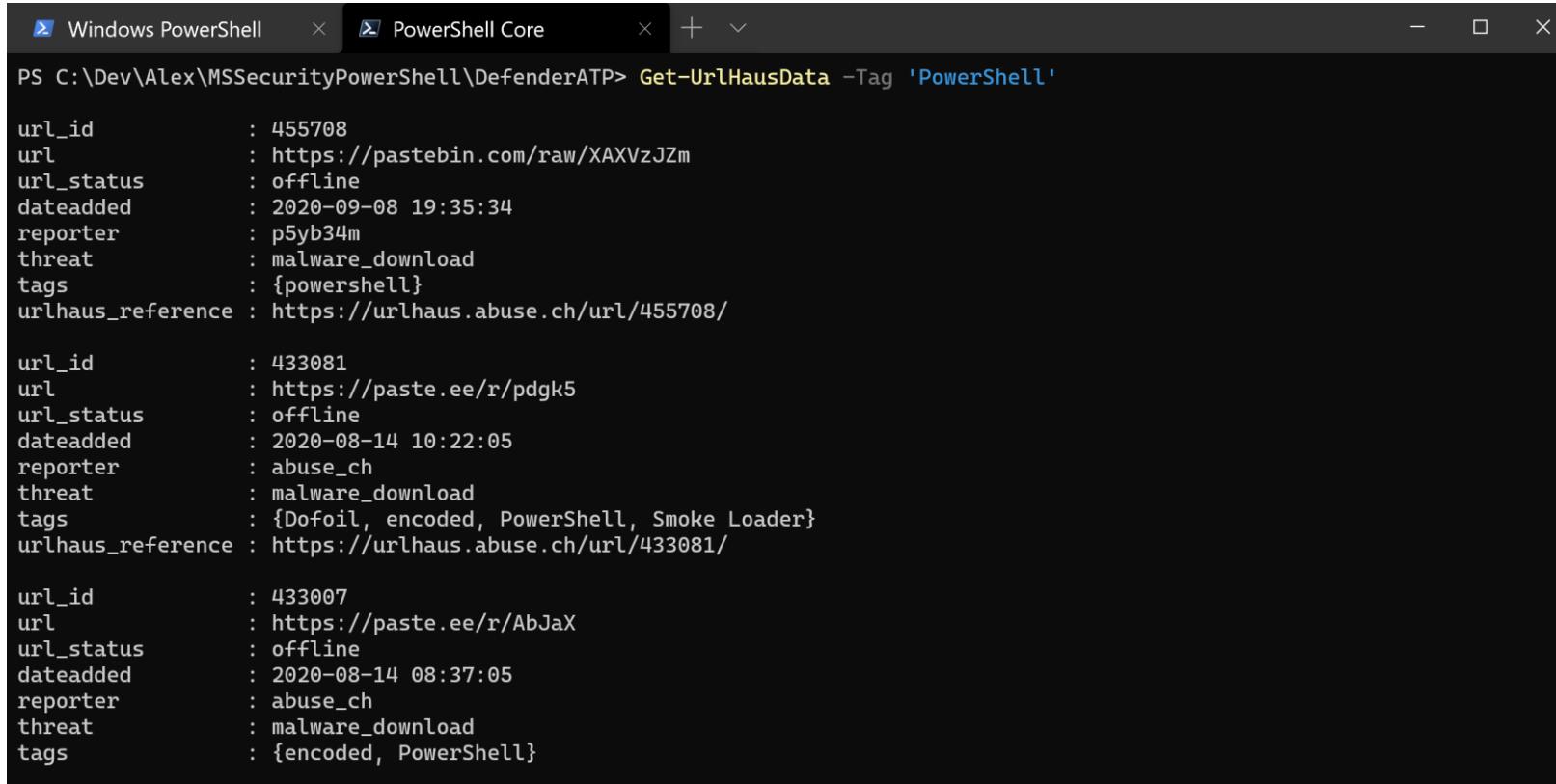
On the right is the 'Group Policy Management' console. It shows the 'Group Policy Objects in contoso.com' list. The table contains 20 entries, each representing a GPO object with details like Name, GPO Status, WMI Filter, Modified date, and Owner. The GPOs include Default Domain Controllers Policy, Default Domain Policy, and numerous MSFT-specific policies for Internet Explorer, Office 365 ProPlus, Windows 10 1903, and Windows Server 1903.

Name	GPO Status	WMI Filter	Modified	Owner
Default Domain Controllers Policy	Enabled	None	10/1/2019 11:41:20 AM	Domain Admin (contoso.com)
Default Domain Policy	Enabled	None	10/1/2019 2:01:24 PM	Domain Admin (contoso.com)
MSFT Internet Explorer 11 - Computer	User configuration setting	None	10/3/2019 6:59:32 PM	Domain Admin (contoso.com)
MSFT Internet Explorer 11 - User	Computer configuration setting	None	10/3/2019 6:59:54 PM	Domain Admin (contoso.com)
MSFT Office 365 ProPlus 1908 - Computer	User configuration setting	None	10/3/2019 6:43:24 PM	Domain Admin (contoso.com)
MSFT Office 365 ProPlus 1908 - Excel DDE B...	Computer configuration setting	None	10/3/2019 6:43:24 PM	Domain Admin (contoso.com)
MSFT Office 365 ProPlus 1908 - Legacy File B...	Computer configuration setting	None	10/3/2019 6:43:23 PM	Domain Admin (contoso.com)
MSFT Office 365 ProPlus 1908 - Remote Mac...	Computer configuration setting	None	10/3/2019 6:43:24 PM	Domain Admin (contoso.com)
MSFT Office 365 ProPlus 1908 - User	Computer configuration setting	None	10/3/2019 6:43:24 PM	Domain Admin (contoso.com)
MSFT Windows 10 1903 - BitLocker	User configuration setting	None	10/3/2019 6:59:46 PM	Domain Admin (contoso.com)
MSFT Windows 10 1903 - Computer	User configuration setting	None	10/3/2019 6:59:48 PM	Domain Admin (contoso.com)
MSFT Windows 10 1903 - User	Computer configuration setting	None	10/3/2019 6:59:34 PM	Domain Admin (contoso.com)
MSFT Windows 10 1903 and Server 1903 - Da...	User configuration setting	None	10/3/2019 6:59:52 PM	Domain Admin (contoso.com)
MSFT Windows 10 1903 and Server 1903 - Do...	User configuration setting	None	10/3/2019 6:59:38 PM	Domain Admin (contoso.com)
MSFT Windows 10 1903 and Server 1903 Me...	User configuration setting	None	10/3/2019 6:59:26 PM	Domain Admin (contoso.com)
MSFT Windows Server 1903 - Domain Controller	User configuration setting	None	10/3/2019 6:59:44 PM	Domain Admin (contoso.com)
MSFT Windows Server 1903 - Domain Controll...	User configuration setting	None	10/3/2019 6:59:40 PM	Domain Admin (contoso.com)
MSFT Windows Server 1903 - Member Server	User configuration setting	None	10/3/2019 6:59:30 PM	Domain Admin (contoso.com)

Managing the Microsoft Security stack with PowerShell – PSURLHAUS

This module provides you with easy-to-use cmdlets to make it easy to interface with the [URLhaus API](#) a project operated by abuse.ch with the purpose of collecting, tracking and sharing malware URLs to support security analysts to protect their network and customers from cyber threats. <https://github.com/alexverboon/PSURLhaus>

Find URLs with PowerShell payload



The screenshot shows a Windows PowerShell window with two tabs: "Windows PowerShell" and "PowerShell Core". The "Windows PowerShell" tab is active and displays the command "Get-UrlHausData -Tag 'PowerShell'" followed by its output. The output lists three URLs, each with various properties like url_id, url, url_status, dateadded, reporter, threat, tags, and urlhaus_reference. The first URL has a tags array containing "powershell". The second URL has a tags array containing "Dofoil, encoded, PowerShell, Smoke Loader". The third URL has a tags array containing "encoded, PowerShell".

```
PS C:\Dev\Alex\MSSecurityPowerShell\DefenderATP> Get-UrlHausData -Tag 'PowerShell'

url_id      : 455708
url         : https://pastebin.com/raw/XAXVzJZm
url_status   : offline
dateadded   : 2020-09-08 19:35:34
reporter    : p5yb34m
threat       : malware_download
tags         : {powershell}
urlhaus_reference : https://urlhaus.abuse.ch/url/455708/

url_id      : 433081
url         : https://paste.ee/r/pdgk5
url_status   : offline
dateadded   : 2020-08-14 10:22:05
reporter    : abuse_ch
threat       : malware_download
tags         : {Dofoil, encoded, PowerShell, Smoke Loader}
urlhaus_reference : https://urlhaus.abuse.ch/url/433081/

url_id      : 433007
url         : https://paste.ee/r/AbJaX
url_status   : offline
dateadded   : 2020-08-14 08:37:05
reporter    : abuse_ch
threat       : malware_download
tags         : {encoded, PowerShell}
```

Managing the Microsoft Security stack with PowerShell – PSURLHAUS

```
Windows PowerShell  PowerShell Core
PS C:\Dev\Alex\MSSecurityPowerShell\DefenderATP> Get-UrlHausData -URL | where url_status -like 'Online'

id          : 456294
urlhaus_reference : https://urlhaus.abuse.ch/url/456294/
url          : http://116.114.95.111:44570/Mozi.m
url_status    : online
host         : 116.114.95.111
date_added   : 2020-09-09 19:19:05 UTC
threat        : malware_download
blacklists    : @{spamhaus dbl=not listed; surbl=not listed}
reporter      : lrz_security
larted       : true
tags          : {elf, Mozi}

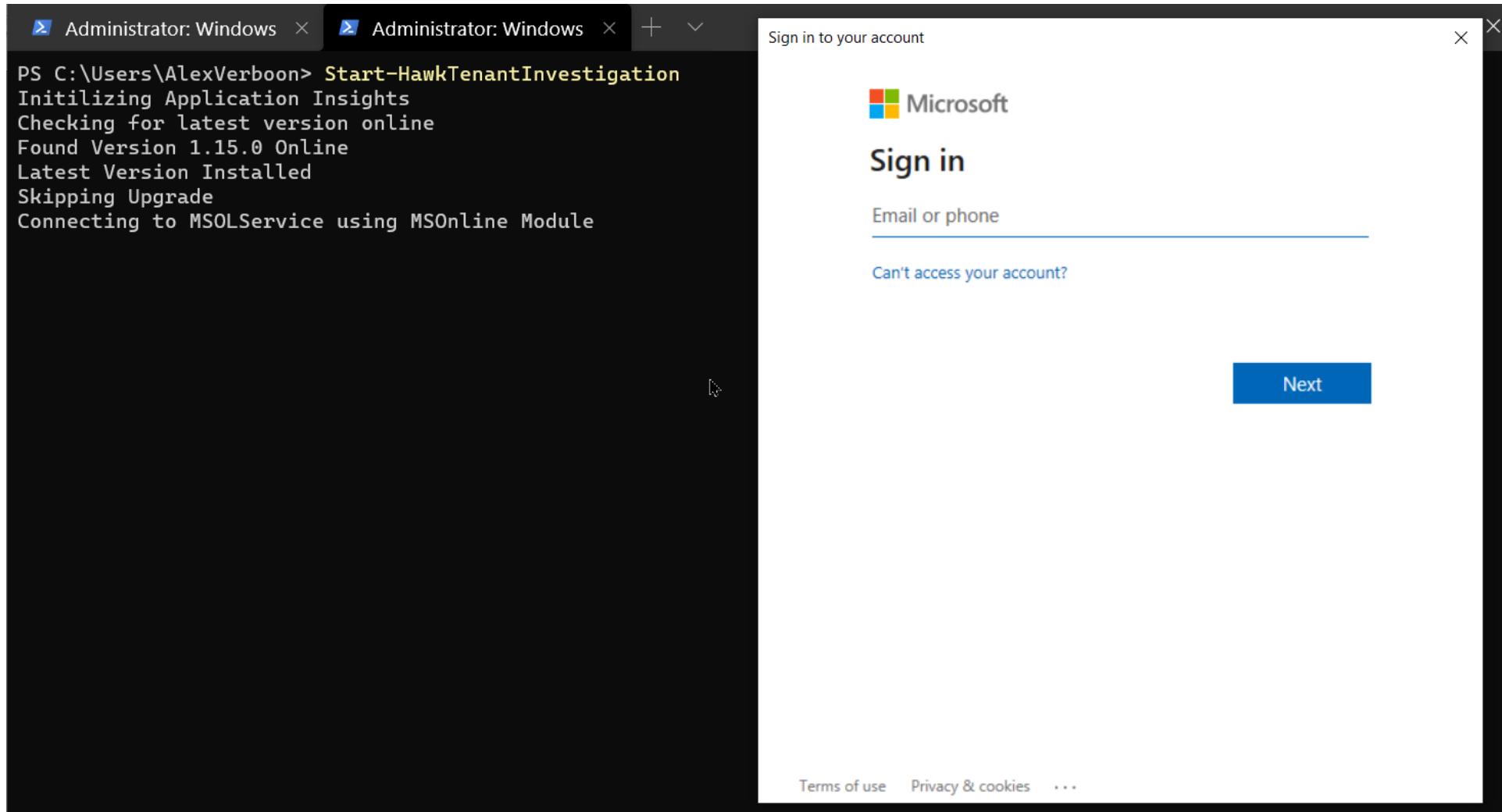
id          : 456292
urlhaus_reference : https://urlhaus.abuse.ch/url/456292/
url          : http://182.117.98.189:47696/Mozi.m
url_status    : online
host         : 182.117.98.189
date_added   : 2020-09-09 19:19:04 UTC
threat        : malware_download
blacklists    : @{spamhaus dbl=not listed; surbl=not listed}
reporter      : lrz_security
larted       : false
tags          : {elf, Mozi}

id          : 456293
```

Managing the Microsoft Security stack with PowerShell – HAWK

<https://github.com/Canthv0/hawk>

Powershell Based tool for gathering information related to O365 intrusions and potential Breaches



Managing the Microsoft Security stack with PowerShell – HAWK

```
Administrator: Windows × Administrator: Windows × + ▾
```

Disclaimer

```
Creating implicit remoting module ...
Getting command information from remote session ... 142 commands received
[oooooooooooooooooooo
00:00:05 remaining.
```

```
First Day of Search Window (1-90, Date, Default 90): 90
Calculating Start Date from current date minus 90 days.
Setting StartDate by Calculation to 06/15/2020 00:00:00

Last Day of search Window (1-90, date, Default Today):
Setting End Date to Today
Setting EndDate by Calculation to 09/14/2020 00:00:00

Advanced Azure AD License NOT Found
Setting up Global Hawk environment variable

[9/13/2020 2:50:01 PM] - Global Variable Configured →
[9/13/2020 2:50:01 PM] - *** Version 1.15.0 ***
[9/13/2020 2:50:01 PM] - @{FilePath=c:\temp\20200913_1449; DaysToLookBack=; StartDate=06/15/2020 00:00:00; EndDate=09/14
/2020 00:00:00; AdvancedAzureLicense=False; WhenCreated=9/13/2020 2:50 PM; EULA=Agreed 09/13/2020 14:49:48}
[9/13/2020 2:50:01 PM] - Starting Tenant Sweep
[9/13/2020 2:50:01 PM] - [ACTION] - Running Get-HawkTenantConfiguration
[9/13/2020 2:50:02 PM] - Not Connected to Exchange Online
[9/13/2020 2:50:02 PM] - Connecting to EXO using CloudConnect Module
```

Managing the Microsoft Security stack with PowerShell – HAWK

```
Administrator: Windows x Administrator: Windows x + - X
[9/13/2020 2:51:00 PM] - Writing Data to c:\temp\20200913_1449\Tenant\XML\RBAC_Changes.xml
[9/13/2020 2:51:00 PM] - Writing Data to c:\temp\20200913_1449\Tenant\RBAC_Changes.csv
[9/13/2020 2:51:00 PM] - [ACTION] - Running Get-HawkTenantAzureAuditLog
[9/13/2020 2:51:01 PM] - [ACTION] - Searching Unified Audit Logs Azure Activities
[9/13/2020 2:51:01 PM] - Searching for Application Activities
[9/13/2020 2:51:01 PM] - Running Unified Audit Log Search
[9/13/2020 2:51:01 PM] - Search-UnifiedAuditLog -RecordType 'AzureActiveDirectory' -Operations 'Add OAuth2PermissionGrant','Consent to application.' -StartDate '6/15/2020' -EndDate '9/14/2020' -SessionCommand ReturnLargeSet -resultsize 100 -sessionid 145101
[9/13/2020 2:51:08 PM] - Retrieved all results.
[9/13/2020 2:51:08 PM] Retrieved:14 Total: 14
[9/13/2020 2:51:08 PM] - ## INVESTIGATE ## - Application Rights Activity found.
[9/13/2020 2:51:08 PM] - ## INVESTIGATE ## - Please review these Azure_Application_Audit.csv to ensure any changes are legitimate.
[9/13/2020 2:51:08 PM] - Appending Data to c:\temp\20200913_1449\Tenant\Azure_Application_Audit.csv
```

A	B	C	D	E	F	G	H	I
Id	Operation	ResultStatus	Workload	ClientIP	Userid	ActorUPN	targetName	env_time
bf006940-951c-4f5f-b510-459bd8564f69	Consent to application.	Success	AzureActiveDirectory	65.52.228.217			Microsoft Graph PowerShell (Preview)	2020-08-16T1
2898debb-a7b2-41ce-8150-b8e8327870c8	Consent to application.	Success	AzureActiveDirectory	40.112.91.238			Microsoft Graph PowerShell (Preview)	2020-08-04T1
eb26ca77-a20a-47f4-9bb8-3a0d0b2bfc1	Consent to application.	Success	AzureActiveDirectory	77.56.162.123			PSMDATP1	2020-07-18T1
2dff712f-2ff4-4176-b9eb-f9d12000e3ee	Consent to application.	Success	AzureActiveDirectory	77.56.162.123			Azure AD Risk Detection API	2020-07-21T1
320246d4-6eaf-4d35-b966-d3f82da3f328	Consent to application.	Success	AzureActiveDirectory	77.56.162.123			Azure AD Risk Detection API	2020-07-21T1
15c131b2-ad3e-4129-8ab0-908ec1a47091	Consent to application.	Success	AzureActiveDirectory	77.56.162.123			ServicePrincipalAlex	2020-07-07T1
2a5f0114-4de9-47b7-8074-3d5df0b6a2ef	Consent to application.	Success	AzureActiveDirectory	77.56.162.123			PSMDATP1	2020-07-18T0
8d15afb8-45f7-4adb-9ee4-4d8a3b167a53	Consent to application.	Success	AzureActiveDirectory	52.138.136.95			WD Antivirus Testground	2020-07-06T0
d390a0b7-c62a-45bc-8c3d-56aa69e6246a	Consent to application.	Success	AzureActiveDirectory	40.112.91.238			Windows Defender ATP for Flow	2020-07-18T1
a1789b73-9719-4abd-b593-53ac717e812e	Consent to application.	Success	AzureActiveDirectory	77.56.162.123			PSMDATP1	2020-07-18T0
9f9aa756-9c11-4b57-88ba-4e23bfd01022	Consent to application.	Success	AzureActiveDirectory	77.56.162.123			PSMDATP1	2020-07-18T0
4c9fd4ce-3946-49bc-ad2d-8f7443bc82b5	Consent to application.	Success	AzureActiveDirectory	77.56.162.123			PSMDATP1	2020-07-18T0
6cddd372e-5bfa-4dd3-9442-7faac6447797	Consent to application.	Success	AzureActiveDirectory	40.127.164.6			WD Antivirus Testground	2020-06-30T1
b5fef836-82e7-42ea-85dc-06e6fadcc6f	Consent to application.	Success	AzureActiveDirectory	65.52.228.217			Azure Notebooks	2020-06-29T2
16								
17								

```
[9/13/2020 2:51:09 PM] - [ACTION] - Running Get-HawkTenantConsentGrants
```

```
[9/13/2020 2:51:09 PM] - Gathering Oauth / Application Grants
```