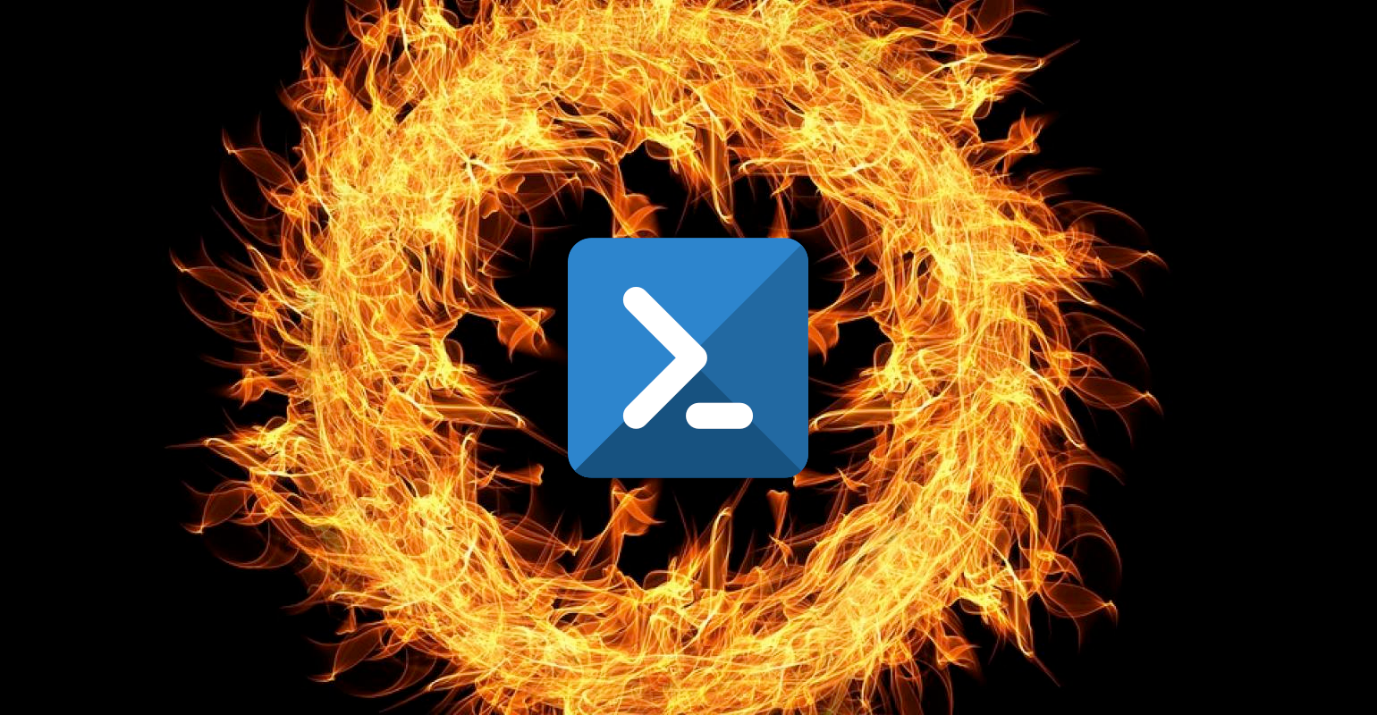
"Hunting Down the Hidden: Detecting the Undetectable with Threat Hunting"



**Here are ten PowerShell XML queries that can be used for threat hunting in Windows environments:**

**1. Detect Suspicious PowerShell Command Line Parameters:**

*```xml*

*<QueryList>*

*<Query Id="0" Path="Security">*

*<Select Path="Security">*

*\*[System[(EventID='4688')]]*

*and*

*\*[EventData[Data[@Name='NewProcessName'] and (Data='C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe')]]*

*and*

*(*

*\*[EventData[Data[@Name='CommandLine'] and (Data contains '-EncodedCommand' or Data contains '-Command' or Data contains '-File')]]*

*or*

*\*[EventData[Data[@Name='CommandLine'] and (Data contains 'powershell.exe -nop -w hidden -c')]]*

*)*

*</Select>*

*</Query>*

*</QueryList>*

*```*

This query looks for PowerShell executions with suspicious command line parameters like `-EncodedCommand`, `-Command`, `-File`, or the presence of specific command-line strings indicating obfuscation.

**2. Detect PowerShell Remote Activity:**

*```xml*

*<QueryList>*

*<Query Id="0" Path="Security">*

*<Select Path="Security">*

*\*[System[(EventID='4688')]]*

*and*

*\*[EventData[Data[@Name='NewProcessName'] and (Data='C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe')]]*

*and*

*\*[EventData[Data[@Name='CommandLine'] and (Data contains 'powershell.exe -noexit -noninteractive -executionpolicy bypass -windowstyle hidden -encodedcommand')]]*

*</Select>*

*</Query>*

*</QueryList>*

*```*

This query identifies PowerShell executions with command line parameters indicating a hidden window and the use of encoded commands, commonly seen in PowerShell-based remote access tools (RATs).

**3. Detect Suspicious PowerShell Module Loading:**

*```xml*

*<QueryList>*

*<Query Id="0" Path="Microsoft-Windows-PowerShell/Operational">*

*<Select Path="Microsoft-Windows-PowerShell/Operational">*

*\*[System[(EventID='4103')]]*

*and*

*\*[EventData[Data[@Name='EngineVersion'] and (Data = '5.1.14409.1005' or Data = '5.1.18362.145')]]*

*and*

*\*[EventData[Data[@Name='ModuleName'] and (Data contains 'invoke-' or Data contains 'mimikatz' or Data contains 'nishang')]]*

*</Select>*

*</Query>*

*</QueryList>*

*```*

This query looks for PowerShell events where specific suspicious modules, such as those commonly used for post-exploitation activities or code injection, are being loaded.

**4. Detect PowerShell Script Block Logging:**

*```xml*

*<QueryList>*

*<Query Id="0" Path="Microsoft-Windows-PowerShell/Operational">*

*<Select Path="Microsoft-Windows-PowerShell/Operational">*

*\*[System[(EventID='4104')]]*

*and*

*\*[EventData[Data[@Name='EngineVersion'] and (Data = '5.1.14409.1005' or Data = '5.1.18362.145')]]*

*and*

*\*[EventData[Data[@Name='ScriptBlockText'] and (Data contains 'downloadfile' or Data contains 'invoke-expression' or Data contains 'net.webclient')]]*

*</Select>*

*</Query>*

*</QueryList>*

*```*

This query detects PowerShell script block logging events that contain suspicious script block text, such as functions commonly used in malicious activities, like file downloads or remote code execution.

**5. Detect PowerShell Command Line Logging:**

*```xml*

*<QueryList>*

*<Query Id="0" Path="Microsoft-Windows-PowerShell/Operational">*

*<Select Path="Microsoft-Windows-PowerShell/Operational">*

*\*[System[(EventID='4106')]]*

*and*

*\*[EventData[Data[@Name='EngineVersion'] and (Data = '5.1.14409.1005' or Data = '5.1.18362.145')]]*

*and*

*\*[EventData[Data[@Name='CommandLine'] and (Data contains 'iex' or Data contains 'downloadstring' or Data contains 'invoke-webrequest')]]*

*</Select>*

*</Query>*

*</QueryList>*

*```*

This query identifies PowerShell command line logging events that contain suspicious command line parameters, indicating potential malicious activities such as executing remote code or downloading files.

**6. Detect PowerShell Empire C2 Framework:**

*```xml*

*<QueryList>*

*<Query Id="0" Path="Security">*

*<Select Path="Security">*

*\*[System[(EventID='4688')]]*

*and*

*\*[EventData[Data[@Name='NewProcessName'] and (Data='C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe')]]*

*and*

*\*[EventData[Data[@Name='CommandLine'] and (Data contains 'powershell.exe -sta -windowstyle hidden -noprofile -encodedcommand')]]*

*</Select>*

*</Query>*

*</QueryList>*

*```*

This query detects PowerShell executions with command line parameters often used by the PowerShell Empire post-exploitation framework.

**7. Detect PowerShell Code Injection:**

*```xml*

*<QueryList>*

*<Query Id="0" Path="Security">*

*<Select Path="Security">*

*\*[System[(EventID='4688')]]*

*and*

*\*[EventData[Data[@Name='NewProcessName'] and (Data='C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe')]]*

*and*

*\*[EventData[Data[@Name='ParentCommandLine'] and (Data contains '-c ')]]*

*</Select>*

*</Query>*

*</QueryList>*

*```*

This query identifies PowerShell executions launched from a parent process with the `-c` parameter, which may indicate code injection or execution of malicious one-liners.

**8. Detect Suspicious PowerShell Outbound Connections:**

*```xml*

*<QueryList>*

*<Query Id="0" Path="Microsoft-Windows-PowerShell/Operational">*

*<Select Path="Microsoft-Windows-PowerShell/Operational">*

*\*[System[(EventID='4104')]]*

*and*

*\*[EventData[Data[@Name='EngineVersion'] and (Data = '5.1.14409.1005' or Data = '5.1.18362.145')]]*

*and*

*(*

*\*[EventData[Data[@Name='ScriptBlockText'] and (Data contains 'invoke-webrequest -uri' or Data contains 'invoke-restmethod -uri')]]*

*or*

*\*[EventData[Data[@Name='ScriptBlockText'] and (Data contains 'net.webclient') and (Data contains '-uri')]]*

*)*

*</Select>*

*</Query>*

*</QueryList>*

*```*

This query looks for PowerShell script block logging events that contain suspicious commands used for making outbound connections, such as `Invoke-WebRequest` or `Invoke-RestMethod` with specific URIs.

**9. Detect PowerShell Event Logging Tampering:**

*```xml*

*<QueryList>*

*<Query Id="0" Path="Microsoft-Windows-PowerShell/Operational">*

*<Select Path="Microsoft-Windows-PowerShell/Operational">*

*\*[System[(EventID='400'))*

*and*

*\*[EventData[Data[@Name='EngineVersion'] and (Data = '5.1.14409.1005' or Data = '5.1.18362.145')]]*

*and*

*\*[EventData[Data[@Name='HostApplication'] and (Data='C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe')]]*

*and*

*(*

*\*[EventData[Data[@Name='HostId'] and (Data = 'System.Management.Automation.Internal.Host.InternalHost')]]*

*or*

*\*[EventData[Data[@Name='HostId'] and (Data = 'ConsoleHost')]]*

*)*

*</Select>*

*</Query>*

*</QueryList>*

*```*

This query detects tampering attempts by checking for missing PowerShell event logs (EventID 400) when the PowerShell host application is `powershell.exe` and the host ID matches internal PowerShell host names.

**10. Detect PowerShell Obfuscated Scripts:**

*```xml*

*<QueryList>*

*<Query Id="0" Path="Microsoft-Windows-PowerShell/Operational">*

*<Select Path="Microsoft-Windows-PowerShell/Operational">*

*\*[System[(EventID='4104')]]*

*and*

*\*[EventData[Data[@Name='EngineVersion'] and (Data = '5.1.14409.1005' or Data = '5.1.18362.145')]]*

*and*

*\*[EventData[Data[@Name='ScriptBlockText'] and (Data contains '-Enc ')]]*

*</Select>*

*</Query>*

*</QueryList>*

*```*

This query detects PowerShell script block logging events that contain obfuscated commands using the `-Enc` parameter, indicating potential malicious activities.

These queries can be used as a starting point for PowerShell threat hunting, but it's important to customize them based on your specific environment and threat intelligence. Additionally, consider combining multiple queries and adding additional filters or indicators to enhance your threat hunting capabilities.