aLERTING AND dETECTION sTRATEGY

Challenge 1

Contents

[Goal 1](#_Toc65524734)

[Categorization 2](#_Toc65524735)

[Strategy Abstract 2](#_Toc65524736)

[Technical Context 2](#_Toc65524737)

[Blind Spots and Assumptions 2](#_Toc65524738)

[False Positives 3](#_Toc65524739)

[Validation 3](#_Toc65524740)

[Priority 3](#_Toc65524741)

[Response 3](#_Toc65524742)

[Additional Resources 3](#_Toc65524743)

[Lab Implementation 3](#_Toc65524744)

# Goal

Detect attempts to bypass PowerShell execution policies. This may indicate that someone is attempting to run malicious code.

# Categorization

Attempting to bypassing PowerShell execution policies is categorized as [Execution / PowerShell](https://attack.mitre.org/techniques/T1059/001/)

# Strategy Abstract

The strategy will function as follows:

* Record PowerShell events using Windows Events capabilities and endpoint detection toolsets.
* Identify and alert on any explicit PowerShell references to known execution policy bypass.
* Record PowerShell script blocks for additional context and analysis.
* Suppress known utility servers.

# Technical Context

PowerShell is a Microsoft scripting language commonly used for management and automation of Windows Servers and endpoints. Legitimate use by Administrators, shared use of PowerShell features, and default installation all make PowerShell an enticing tool for malicious actors.

Additionally, PowerShell provides an attacker the ability to run malicious code in memory and run obfuscated code, making detection more difficult.

PowerShell v5 provides robust logging capabilities that allow monitoring of script blocks, command transcription, and module logging. While many logging functionalities are backported to v4, v5 is the recommended implementation. Systems with previous versions of PowerShell installed are susceptible to attacks that circumvent enhanced logging utilizing PowerShell downgrade attacks.

PowerShell provides a default execution policy of “Restricted”. The execution policy was not created to be a security control, but it does determine whether configuration files can be loaded and scripts can be run. You can view the current execution policy by running “Get-ExecutionPolicy” in PowerShell.

# Blind Spots and Assumptions

This strategy relies on the following assumptions:

* Module logging and Script Block Logging are enabled.
* Additional analysis – PowerShell Transcription is enabled.
* Module logging is not filtered on specific modules to limit visibility.
* Logs are configured correctly to forward to SIEM.
* SIEM is aggregating logs to a central index.
* An alert is created within the SIEM for execution bypass attempts.

A blind spot will occur if the following scenarios are met:

* Endpoint event logs are not forwarded to SIEM.
* A utility server that has been whitelisted is exploited.

# False Positives

* A legitimate installer or similar vendor-provided tool utilize common execution policy bypasses.
* An administrator utilizes an execution policy bypass during legitimate work.

# Validation

Validation for this strategy can be confirmed with the following:

Within a command prompt window:

* Echo Write-Host "This is a test for SpecterOps" | PowerShell.exe
* PowerShell.exe -ExecutionPolicy Bypass -command "Write-host 'This is a test for SpecterOps'"

Within a PowerShell window:

* Set-ExecutionPolicy Bypass CurrentUser

# Priority

In a PowerShell shop, events are to be categorized as a medium priority event. In a business that does not leverage PowerShell, additional priority is to be assigned. Combined correlation with additional events constitutes further priority. i.e., Invoke-WebRequest and variations of -EncodedCommand.

# Response

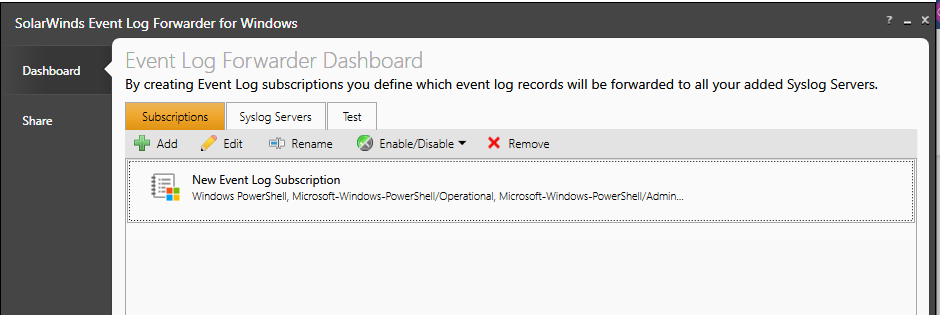
* Review the user who spawned the process.
* Review execution behavior in endpoint tools.
  + Identify parent and child processes.
  + Identify network connections.
  + Investigate system or file interactions.
* Review script block logs and transcription logs to verify intent.
* Follow up with source user if expected to be legitimate.

# Additional Resources

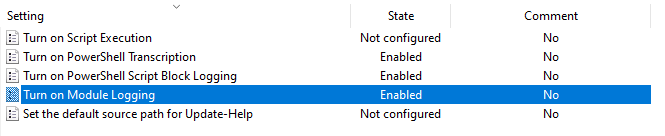
* [Enabling PowerShell logging](https://www.fireeye.com/blog/threat-research/2016/02/greater_visibilityt.html)
* [Downgrade Attacks](https://www.leeholmes.com/blog/2017/03/17/detecting-and-preventing-powershell-downgrade-attacks/)
* [PowerShell Threat Detection](https://redcanary.com/threat-detection-report/techniques/powershell/)
* [PowerShell Execution Bypass](https://blog.netspi.com/15-ways-to-bypass-the-powershell-execution-policy/)

# Lab Implementation

Endpoint - Event log forwarding



GPO Administrative Templates -> Windows Components -> Windows PowerShell



Splunk

