PowerShell – Cybersecurity Perspective

Malicious PowerShell scripts are predominantly used as downloaders (such as Office macros) during the incursion phase. The second most common use is during the lateral movement phase, allowing a threat actor to execute code on a remote computer when spreading inside the network.

PowerShell can be run in various ways:

- from PowerShell console – the most common one,
- from within a C#/.NET applications,
- from command-line,
- from macros in MS Office documents,
- from various scripts including VBScripts.

All internal legitimately used PowerShell scripts should be signed and all unsigned scripts should be blocked through the ex ecution policy. While there are simple ways to bypass the execution policy, enabling it makes infection more difficult. The security team should be able to monitor for any attempt to bypass the execution policy and follow up on it.

For the detection of attacks involving Powershell, it is recommend to establish a baseline of normal activity in an environment. Deviations from the baseline may serve as an indication of attacker activity. It is recommend that organizations formulate a PowerShell monitoring strategy by first assessing and enumerating :

- Which servers/server groups are administered via PowerShell remoting?
- What are the source hostnames of systems used to administer via Powershell?
- What are the names and common directories used for legitimate PowerShell scripts within the environment?
- Are any systems configured to automatically load and execute PowerShell scripts for administration purposes?

Windows PowerShell event log entries indicating the start and stop of PowerShell activity

- Event ID 400: “Engine state is changed from None to Available”, upon the start of any local or remote PowerShell activity.
- Event ID 600: referencing “WSMan” (e.g. “Provider WSMan Is Started”), indicating the onset of PowerShell remoting activity on both source and destination systems.
- Event ID 403: “Engine state is changed from Available to Stopped”, upon the end of the PowerShell activity.
- Event ID 40961: “PowerShell console is starting up”
- Event ID 4100: “Error Message = File (path to)test.ps1 cannot be loaded. . . ”

System event log entries indicating a configuration change to the Windows Remote Management service:

- Event ID 7040: “The start type of the Windows Remote Management (WS-Management) service was changed from [disabled / demand start] to auto start.” – recorded when PowerShell remoting is enabled.
- Event ID 10148: “The WinRM service is listening for WS-Management requests” –recorded upon reboot on systems where remoting has been enabled.

Security event log entries indicating the execution of the PowerShell console or interpreter:

- Event ID 4688: “A new process has been created” – includes account name, domain, and executable name in the event message.

WinRM Operational event log entries indicating authentication prior to PowerShell remoting on an accessed system:

- Event ID 169: “User [DOMAIN\Account] authenticated successfully using [authentication_protocol]”

AppLocker event log entries recording the local execution of PowerShell scripts:

- Event ID 8005: “[script_path] was allowed to run”
- Event ID 8006: “[script_path] was allowed to run but would have been prevented from running if the AppLocker policy were enforced”

Sysmon Event Sample Queries:

Powershell Started from Suspicious Processes. Objective: Monitor browser and MS Office processes for cmd or PowerShell as child process on workstations.

- *process.parent.name: (winword.exe OR outlook.exe OR excel.exe OR powrpnt.exe) AND process.name: (powershell.exe)*

Detecting Long PowerShell Command. Objective: Detect long PowerShell commands that most probably will include obfuscated malicious code by length of command.

- *process.name: (powershell.exe) AND command.line_length: >1000*

Search for Suspicious Strings. Objective: This query searches for the presence of Invoke-Expression or IEX or Download strings

- *process.name: (powershell.exe) AND process.command_line: (invoke* OR iex OR download*)*