Event Triggered Execution: PowerShell Profile

Other sub-techniques of Event Triggered Execution (17)

Adversaries may gain persistence and elevate privileges by executing malicious content triggered by PowerShell profiles. A PowerShell profile (`profile.ps1`) is a script that runs when PowerShell starts and can be used as a logon script to customize user environments.

PowerShell supports several profiles depending on the user or host program. For example, there can be different profiles for PowerShell host programs such as the PowerShell console, PowerShell ISE or Visual Studio Code. An administrator can also configure a profile that applies to all users and host programs on the local computer. [1]

Adversaries may modify these profiles to include arbitrary commands, functions, modules, and/or PowerShell drives to gain persistence. Every time a user opens a PowerShell session the modified script will be executed unless the `-NoProfile` flag is used when it is launched. [2]

An adversary may also be able to escalate privileges if a script in a PowerShell profile is loaded and executed by an account with higher privileges, such as a domain administrator. [3]

ID: T1546.013

Sub-technique of: T1546

Tactics: Privilege Escalation, Persistence

Platforms: Windows

Permissions Required: Administrator, User

Contributors: Allen DeRyke, ICE; Matt Green, @mgreen27

Version: 1.1

Created: 24 January 2020

Last Modified: 20 October 2023

Version Permalink

Procedure Examples

ID	Name	Description
G0010	Turla	Turla has used PowerShell profiles to maintain persistence on an infected machine.[2]

Mitigations

ID	Mitigation	Description
M1045	Code Signing	Enforce execution of only signed PowerShell scripts. Sign profiles to avoid them from being modified.
M1022	Restrict File and Directory Permissions	Making PowerShell profiles immutable and only changeable by certain administrators will limit the ability for adversaries to easily create user level persistence.
M1054	Software Configuration	Avoid PowerShell profiles if not needed. Use the -No Profile flag with when executing PowerShell scripts remotely to prevent local profiles and scripts from being executed.

Detection

ID	Data Source	Data Component	Detects
DS0017	Command	Command Execution	Monitor abnormal PowerShell commands, unusual loading of PowerShell drives or modules.
DS0022	File	File Creation	Locations where `profile.ps1` can be stored should be monitored for new profiles. [4] Example profile locations include:* `$PsHome\Profile.ps1` * `$PsHome\Microsoft.{HostProgram}_profile.ps1` * `$Home\My Documents\PowerShell\Profile.ps1` * `$Home\My Documents\PowerShell\Microsoft.{HostProgram}_profile.ps1`
		File Modification	Locations where `profile.ps1` can be stored should be monitored for modifications. [4] Example profile locations include:* `$PsHome\Profile.ps1` * `$PsHome\Microsoft.{HostProgram}_profile.ps1` * `$Home\My Documents\PowerShell\Profile.ps1` * `$Home\My Documents\PowerShell\Microsoft.{HostProgram}_profile.ps1`
DS0009	Process	Process Creation	Monitor newly executed processes that may gain persistence and elevate privileges by executing malicious content triggered by PowerShell profiles.

References

1. Microsoft. (2017, November 29). About Profiles. Retrieved June 14, 2019.
2. Faou, M. and Dumont R.. (2019, May 29). A dive into Turla PowerShell usage. Retrieved June 14, 2019.
3. DeRyke, A.. (2019, June 7). Lab Notes: Persistence and Privilege Elevation using the Powershell Profile. Retrieved July 8, 2019.
4. Malware Archaeology. (2016, June). WINDOWS POWERSHELL LOGGING CHEAT SHEET - Win 7/Win 2008 or later. Retrieved June 24, 2016.