Keeping PowerShell: Security Measures to Use and Embrace

Cybersecurity authorities from the United States, New Zealand, and the United Kingdom recommend proper configuration and monitoring of PowerShell, as opposed to removing or disabling PowerShell entirely. This will provide benefits from the security capabilities PowerShell can enable while reducing the likelihood of malicious actors using it undetected after gaining access into victim networks. The following recommendations will help defenders detect and prevent abuse by malicious cyber actors, while enabling legitimate use by administrators and defenders.

This Cybersecurity Information Sheet from the National Security Agency (NSA), the Cybersecurity and Infrastructure Security Agency (CISA), the New Zealand National Cyber Security Centre (NZ NCSC), and the United Kingdom National Cyber Security Centre (NCSC-UK) provides details on using PowerShell® and its security measures.

PowerShell® is a scripting language and command line tool included with Microsoft Windows®. Similar to Bash for open-source operating systems (e.g., Linux®), PowerShell extends the user experience as an interface into the operating system. PowerShell was introduced in Windows Vista® and has evolved with each Windows version. PowerShell can help defenders manage the Windows operating system [1], by:

- Enabling forensics efforts,
- Improving incident response, and
- Allowing automation of common or repetitive tasks.

In Microsoft's cloud platform Azure®, PowerShell can help to manage Azure resources, permitting administrators and defenders to build automated tools and security measures. However, the extensibility, ease of use, and availability of PowerShell also presents an opportunity for malicious cyber actors. Many publicly-acknowledged cyber intrusions, including those by ransomware actors, have used PowerShell as a post-exploitation tool [2], [3], [4]. This technique is not new, as malicious actors often find ways to target or use legitimate system software.

The authors' recommendations mitigate cyber threats without obstructing PowerShell's functionality, which aligns to Microsoft's guidance on maintaining operational

U/OO/166015 | PP-22-0923 | JUN 2022 Ver 1.0

PowerShell use [5]. Blocking PowerShell hinders defensive capabilities that current versions of PowerShell can provide, and prevents components of the Windows operating system from running properly [5]. Recent versions of PowerShell with improved capabilities and options can assist defenders in countering abuse of PowerShell. The Australian Cyber Security Centre (ACSC) has also offered comprehensive configuration guidance [6] on securing PowerShell.

Current defensive landscape for PowerShell

PowerShell 7.2 is the latest release, though an earlier version, 5.1, is included with Windows 10+ [7], [8]. Version 7.2 is managed and open sourced by Microsoft [9] [10][1]. In Windows 10+ with proper configuration, PowerShell 7.2 can fully integrate with and access all components created for version 5.1, allowing for continued use of existing scripts, modules, and commands. Malicious PowerShell use prior to version 5.0 motivated public efforts to detect those targeted PowerShell actions [11]. Recent PowerShell versions (see ***Table 1***) include enhanced security measures, such as the prevention, detection, and authentication capabilities [12] detailed in the following sections. The authors recommend explicitly disabling and uninstalling the deprecated second version of PowerShell (i.e., Version 2) on Windows 10+ to defend against bypasses of defenses described below [13].

PowerShell methods to reduce abuse

Built-in Windows security features available in PowerShell can reduce abuse by cyber actors. The authors recommend using these capabilities where feasible.

Credential protection during PowerShell remoting

PowerShell remoting is a Windows capability that enables administrators, cybersecurity analysts, and users to remotely execute commands on Windows hosts [14]. Windows Remote Management (WinRM) is the underlying protocol used by PowerShell remoting and uses Kerberos or New Technology LAN Manager (NTLM) as the default authentication protocols. These authentication protocols do not send the actual credentials to remote hosts, avoiding direct exposure of credentials and risk of theft through revealed credentials [14].

[1] Recent PowerShell versions can also function on other operating systems (e.g., Linux, MacOS®) using a limited set of modules. The authors make no recommendation on the use of PowerShell with other operating systems.

Network protection of PowerShell remoting

Remote connections can be used for powerful remote management capabilities, so Windows Firewall rules on endpoints should be configured appropriately to control permitted connections. The client and server editions of Windows include PowerShell remoting, with this capability enabled by default on Windows servers beginning with Windows 2012 R2 [14]. Access to endpoints with PowerShell remoting requires the requesting user account to have administrative privileges at the destination by default. Enabling PowerShell remoting on private networks will introduce a Windows Firewall rule to accept all connections [14]. The permission requirement and Windows Firewall rules are customizable for restricting connections to only trusted endpoints and networks to reduce lateral movement opportunities. Organizations can implement these rules to harden network security where feasible.

Antimalware Scan Interface (AMSI) integration

The Antimalware Scan Interface feature, first available on Windows 10, is integrated into different Windows components. It supports scanning of in-memory and dynamic file contents using an anti-virus product registered with Windows and exposes an interface for applications to scan potentially malicious content [15], [16], [17], [18]. Built-in scripting languages (e.g., PowerShell, VBScript, and JScript®) use AMSI so that scripts are scanned by registered and supported anti-virus software. This feature requires AMSI-aware anti-virus products, such as Windows Defender, McAfee, and Symantec [19], [20], [21].

Constrained PowerShell with Application Control

Configuring AppLocker or Windows Defender Application Control (WDAC) to block actions on a Windows host will cause PowerShell to operate in Constrained Language Mode (CLM), restricting PowerShell operations unless allowed by administrator defined policies. This feature corrects a shortcoming of AppLocker script enforcement that blocks PowerShell commands in a script, but allows the same commands interactively entered into the PowerShell command console. Proper configuration of WDAC or AppLocker [22], [23], [24] on Windows 10+ helps to prevent a malicious actor from gaining full control over a PowerShell session and the host [25]. Controlling the origin and execution of scripts and modules enables opportunities to enhance security requirements and code signing pipelines within organizations. Signing requirements are

also enforceable through PowerShell's safety feature called Execution Policy [26]. However, Execution Policy does not restrict execution of all PowerShell content [26].

PowerShell methods to detect abuse

Logging of PowerShell activities can record when cyber threats leverage PowerShell, and continuous monitoring of PowerShell logs can detect and alert on potential abuses. Deep Script Block Logging, Module Logging, and Over-the-Shoulder transcription are disabled by default. The authors recommend enabling the capabilities where feasible.

Deep Script Block Logging (DSBL) and module logging

Deep Script Block Logging records each PowerShell command in the Windows Event Log, enabling additional analysis on centralized storage and analysis platforms. DSBL records even hidden malicious PowerShell activities and the commands that are executed, such as command invocations, and portions of scripts. Similarly, module logging captures the pipeline execution details of PowerShell, with the goal to record PowerShell actions. Although full details and output may not be recorded, these module logs and event logs prevent PowerShell commands from being obscured (e.g., obfuscated or encrypted) from defenders.

Over-the-Shoulder (OTS) transcription

The capability to record all activities executed within PowerShell 5 can be applied in Windows 7 and later, for both in-the-moment record keeping and restricted security tracking. OTS records every PowerShell input and output, whether functional or not, to enable defenders to decipher intended actions. PowerShell 5.0 expanded the scope of transcription, which is manageable via Group Policy for enterprise-wide configuration.

PowerShell procedures to provide authentication

Multiple authentication methods in PowerShell permit use on non-Windows devices.

Remoting over SSH

PowerShell 7 [27] permits remote connections over Secure Shell (SSH) in addition to supporting WinRM connections. This allows for public key authentication and makes remote management through PowerShell of machines convenient and secure. New SSH remoting capability in PowerShell can establish remote connections without requiring the use of Hypertext Transfer Protocol Secure (HTTPS) with Secure Sockets

Layer/Transport Layer Security (SSL/TLS) certificates. PowerShell over SSH does not require Trusted Hosts [28] as when remoting over WinRM outside of a domain. This allows for secure remote management over SSH without a password for all commands and connections, and enables PowerShell remoting between Windows and Linux hosts.

PowerShell version and security feature availability

The following table lists features available when using a specific version of PowerShell:

Table 1: Available PowerShell features per version and operating system

PowerShell Version (v)	Operating System	AMSI	CLM	CLM with AppLocker and WDAC	DSBL	Over-the-Shoulder Transcription	Module Logging	SSH Remoting
v3	Windows 8		✓				✓	
v4	Windows 8.1		✓			✓	✓	
v5	Windows 10	✓	✓	✓	✓	✓	✓	
v5	Windows 11	✓	✓	✓	✓	✓	✓	
v7	Windows 10	✓	✓	✓	✓	✓	✓	✓
v7	Windows 11	✓	✓	✓	✓	✓	✓	✓
v7	Linux		✓		✓	✓	✓	✓

Conclusion

PowerShell is essential to secure the Windows operating system, especially since newer versions have resolved previous limitations and concerns through updates and enhancements. Removing or improperly restricting PowerShell would prevent administrators and defenders from utilizing PowerShell to assist with system maintenance, forensics, automation, and security. PowerShell, along with its administrative abilities and security measures, should be managed properly and adopted.

Works cited

[1] Microsoft Corporation, "Windows PowerShell (Monad) Has Arrived," [Online]. Available: https://devblogs.microsoft.com/powershell/windows-powershell-monad-has-arrived/.

[2] Trend Micro Inc., "Word & Excel files Infected Using Windows PowerShell," [Online]. Available: https://www.trendmicro.com/en_us/research/14/c/word-and-excel-files-infected-using-windows-powershell.html.

[3] Trend Micro Inc., "Security 101: The Rise of Fileless Threats that Abuse PowerShell," [Online]. Available: https://www.trendmicro.com/vinfo/us/security/news/security-technology/security-101-the-rise-of-fileless-threats-that-abuse-powershell.

[4] Cybersecurity and Infrastructure Security Agency, "Cybersecurity & Infrastructure Security Agency (CISA) FiveHands Ransomware Analysis Report (AR21-126A)," [Online]. Available: https://us-cert.cisa.gov/ncas/analysis-reports/ar21-126a.

[5] Microsoft Corporation, "Defending Against PowerShell Attacks," [Online]. Available: https://devblogs.microsoft.com/powershell/defending-against-powershell-attacks.

[6] Australian Cyber Security Centre (ACSC), "Securing PowerShell," [Online]. Available: https://www.cyber.gov.au/acsc/view-all-content/publications/securing-powershell-enterprise.

[7] Microsoft Corporation, "What's New in PowerShell 7.0," [Online]. Available: https://docs.microsoft.com/en-us/powershell/scripting/whats-new/what-s-new-in-powershell-70?view=powershell-7.2.

[8] Microsoft Corporation, "General Availability of PowerShell 7.2," [Online]. Available: https://devblogs.microsoft.com/powershell/general-availability-of-powershell-7-2/.

[9] Microsoft Corporation, "PowerShell on Linux and Open Source!," [Online]. Available: https://devblogs.microsoft.com/powershell/powershell-on-linux-and-open-source-2/.

[10] Microsoft Corporation, "Microsoft's PowerShell GitHub Project," [Online]. Available: https://github.com/powershell/powershell.

[11] Mandiant, "Mandiant – Investigating PowerShell Attacks," [Online]. Available: https://www.blackhat.com/docs/us-14/materials/us-14-Kazanciyan-Investigating-Powershell-Attacks-WP.pdf.

[12] Microsoft Corporation, "PowerShell ♥ the Blue team," [Online]. Available: https://devblogs.microsoft.com/powershell/powershell-the-blue-team/.

[13] Microsoft Corporation, "Windows PowerShell 2.0 Deprecation," [Online]. Available: https://devblogs.microsoft.com/powershell/windows-powershell-2-0-deprecation/.

[14] Microsoft Corporation, "Security Considerations for PowerShell Remoting using WinRM," [Online]. Available: https://docs.microsoft.com/en-us/powershell/scripting/learn/remoting/winrmsecurity?view=?powershell-7.2.

[15] Microsoft Corporation, "Windows 10 to offer application developers new malware defenses," [Online]. Available: https://www.microsoft.com/security/blog/2015/06/09/windows-10-to-offer-application-developers-new-malware-defenses/?source=mmpc.

[16] Microsoft Corporation, "How Antimalware Scan Interface (AMSI) helps you defend against malware," [Online]. Available: https://docs.microsoft.com/en-us/windows/win32/amsi/how-amsi-helps.

[17] Microsoft Corporation, "More about AMSI integration with Exchange Server," [Online]. Available: https://techcommunity.microsoft.com/t5/exchange-team-blog/more-about-amsi-integration-with-exchange-server/ba-p/2572371.

[18] Microsoft Corporation, "XLM + AMSI: New runtime defense against Excel 4.0 macro malware," [Online]. Available: https://www.microsoft.com/security/blog/2021/03/03/xlm-amsi-new-runtime-defense-against-excel-4-0-macro-malware/.

[19] BlackHat USA 2016, "AMSI Win10 Stop Script-Based Attacks," [Online]. Available: https://www.blackhat.com/docs/us-16/materials/us-16-Mittal-AMSI-How-Windows-10-Plans-To-Stop-Script-Based-Attacks-And-How-Well-It-Does-It.pdf.

[20] McAfee, LLC, "McAfee AMSI Integration Protects Against Malicious Scripts," [Online]. Available: https://www.mcafee.com/blogs/other-blogs/mcafee-labs/mcafee-amsi-integration-protects-against-malicious-scripts/.

[21] Broadcom, "Symantec Endpoint Protection Installation and Administration Guide," [Online]. Available: https://techdocs.broadcom.com/us/en/symantec-security-software/endpoint-security-and-management/endpoint-protection/all/release-notes/Whats-new-for-Symantec-Endpoint-Protection-14-3-.html.

[22] Microsoft Corporation, "Windows Defender Application Control - WDAC," [Online]. Available: https://docs.microsoft.com/en-us/hololens/windows-defender-application-control-wdac.

[23] Microsoft Corporation, "AppLocker," [Online]. Available: https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/applocker/applocker-overview.

[24] NSAcyber GitHub, "AppLocker-Guidance," [Online]. Available: https://github.com/nsacyber/AppLocker-Guidance.

[25] Microsoft Corporation, "about_Signing," [Online]. Available: https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_signing?view=powershell-7.2.

[26] Microsoft Corporation, "about_Execution_Policies," [Online]. Available: https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_execution_policies?view=powershell-7.2.

[27] Microsoft Corporation, "Migrating from Windows PowerShell 5.1 to PowerShell 7," [Online]. Available: https://docs.microsoft.com/en-us/powershell/scripting/whats-new/migrating-from-windows-powershell-51-to-powershell-7?view=powershell-7.2.

[28] Microsoft Corporation, "How to add a computer to the trusted hosts list," [Online]. Available: https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_remote_troubleshooting?view=powershell-7.2#how-to-add-a-computer-to-the-trusted-hosts-list.

Disclaimer of endorsement

Trademarks

Purpose

This document was developed in furtherance of NSA's cybersecurity missions, including its responsibilities to identify and disseminate threats to National Security Systems, Department of Defense, and Defense Industrial Base information systems, and to develop and issue cybersecurity specifications and mitigations. This may be shared broadly to reach all appropriate stakeholders.

Contact

Client requirements or general cybersecurity inquiries: Cybersecurity_Requests@nsa.gov
Defense Industrial Base inquiries and cybersecurity services: DIB_Defense@cyber.nsa.gov
Media inquiries: 443-634-0721, MediaRelations@nsa.gov

U.S. organizations: report incidents and anomalous activity to CISA 24/7 Operations Center at report@cisa.gov
New Zealand organizations: report cyber security incidents to incidents@ncsc.govt.nz or call 04 498 7654
United Kingdom organizations: report a significant cyber security incident: ncsc.gov.uk/report-an-incident (monitored 24 hours) or, for urgent assistance, call 03000 200 973