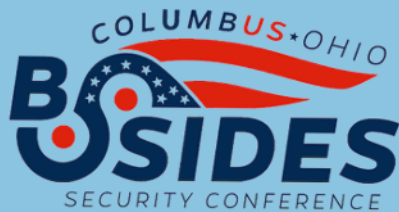


Zoom background for recording



PowerShell pew pew pew:
skilz 4 blue team

Mr. Ashley McGlone

Tanium, TAM
ashley.mcglone@tanium.com
@GoateePFE

Abstract

Tired of the red team taking your lunch money with commodity PowerShell-based attacks? Learn how to track every move they make with Windows features like transcription, module logging, script block logging, and a few other tricks. Do the same in the latest PowerShell Core on MacOS and Linux. Find out about a few gotchas before implementing company-wide. Learn it directly from a former Microsoft insider. Leave with free techniques you can use today.

Bio

Ashley McGlone is a former Microsoft Premier Field Engineer and now a Technical Account Manager at Tanium. He has a familiar face (or goatee) in the PowerShell community as a blogger, tweeter, and speaker. You can find his content on YouTube and TechNet. While at Microsoft Ashley created and delivered PowerShell training to customers around the world. Now at Tanium he is helping companies catch PowerShell malware at scale. Previous attendees of his sessions have said that he is both "informative and entertaining". Ashley's goal is to help people use PowerShell securely in the enterprise.

How we got here

Windows PowerShell Policies

PowerShell Core Policies on Mac, Linux, and Windows

Resources

Learning Objectives

A Brief History of PowerShell Security

2003

Monad announced at Professional Developers Conference

<https://en.wikipedia.org/wiki/PowerShell>

2006

PowerShell 1.0 released

<https://en.wikipedia.org/wiki/PowerShell>

```
PowerShell.exe  
-ExecutionPolicy Bypass  
-File c:\temp\invoke-evil.ps1
```

2008

Blog Post: PowerShell Security Guiding Principles

<https://blogs.msdn.microsoft.com/powershell/2008/09/30/powershells-security-guiding-principles/>

2013

Blog Post: PowerShell Security Best Practices

<https://devblogs.microsoft.com/powershell/powershell-security-best-practices/>

PowerShell version 5 (included in Windows 10, and also available for earlier operating systems through the Windows Management Framework) has made significant strides in making sure that the Blue Team has the information it needs to answer these questions.

KB 3000850 for PowerShell v4 on Windows 8.1 also includes many of these features, as called out below.

2015

Blog Post: PowerShell ♥ the Blue Team

<https://devblogs.microsoft.com/powershell/powershell-the-blue-team/>



2015

PowerShell Empire premiered at BSidesLV

<https://www.youtube.com/watch?v=Pq9t59w0mUI>

Engine	Event Logging	Transcription	Dynamic Evaluation Logging	Encrypted Logging	Application Whitelisting	Antimalware Integration	Local Sandboxing	Remote Sandboxing	Untrusted Input Tracking
Bash	No**	No*	No	No	Yes	No	No*	Yes	No
CMD / BAT	No	No	No	No	Yes	No	No	No	No
IsScript	No	No	No	No	Yes	Yes	No	No	No
LUA	No	No	No	No	No	No	No*	Yes	Yes
Perl	No	No	No	No	No	No	No*	Yes	Yes
PHP	No	No	No	No	No	No	No*	Yes	Yes
PowerShell	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No**
Python	No	No	No	No	No	No	No	No	No**
Ruby	No	No	No	No	No	No	No**	No**	Yes
sh	No**	No*	No	No	No	No	No*	Yes	No
T-SQL	Yes	Yes	Yes	No	No	No	No**	No**	No
VBScript	No	No	No	No	Yes	Yes	No	No	No
zsh	No**	No*	No	No	No	No	No*	Yes	No

* Feature exists, but cannot enforce by policy
 ** Experiments exist

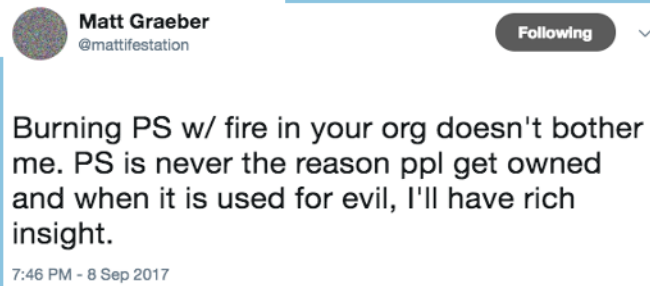
2017

Blog Post: A Comparison of Shell and Scripting Language Security

<https://blogs.msdn.microsoft.com/powershell/2017/04/10/a-comparison-of-shell-and-scripting-language-security/>

2017

Matt Graeber is moving on



<https://twitter.com/mattifestation/status/906315527609507840>

2018

PowerShell Core 6.0 released with cross-platform security built-in

```
{
  "Microsoft.PowerShell.ExecutionPolicy": "RemoteSigned",
  "PowerShellPolicies": {
    "ScriptExecution": {
      "ExecutionPolicy": "RemoteSigned",
      "EnableScripts": true
    },
    "ScriptBlockLogging": {
      "EnableScriptBlockInvocationLogging": true,
      "EnableScriptBlockLogging": true
    },
    "ModuleLogging": {
      "EnableModuleLogging": false,
      "ModuleNames": [
        "PSReadline",
        "PowerShellGet"
      ]
    },
    "ProtectedEventLogging": {
      "EnableProtectedEventLogging": false,
      "EncryptionCertificate": {
        "Joe"
      }
    },
    "Transcription": {
      "EnableTranscripting": true,
      "EnableInvocationHeader": true,
      "OutputDirectory": "F:\\tmp\\new"
    },
    "UpdatableHelp": {
      "DefaultSourcePath": "f:\\temp"
    },
    "ConsoleSessionConfiguration": {
      "EnableConsoleSessionConfiguration": false,
      "ConsoleSessionConfigurationName": "name"
    }
  },
  "LogLevel": "verbose"
}
```

https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_logging_non-windows?view=powershell-7.1#configuring-logging-on-a-non-windows-system


The file powershell.config.json is a JSON formatted file residing in the PowerShell \$PSHOME directory.

2019

PowerShell Empire is dead. New tools emerge.




<https://twitter.com/xorrior/status/1156626181107736576>




Ransomware, "wiper" malware attacks have more than doubled, IBM team says

X-Forensics response to cases of destructive malware increased 200% ... **PowerShell scripts** are still heavily used by ransomware attacks to ...

1 month ago







FBI Warns of Banking Trojan That Eludes Antivirus as Fileless ...

Fileless malware is a growing threat to the banking industry requiring ... or using legitimate system administration tools such as **PowerShell**.

4 weeks ago







Crypto-Mining Malware Attacks up 29% in Q1: McAfee Report

Crypto-mining malware campaigns climbed 29 percent from Q4 2018 ... By **using PowerShell**, vulnerabilities in servers were exploited to mine ...

5 days ago

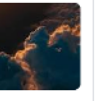





Cloud Atlas threat group updates weaponry with polymorphic malware

*This updated version is carried out in order to make the malware invisible to **this PowerShell-based** module is also able to receive further ...

3 weeks ago






FIN6 Switches Up PoS Tactics to Target E-Commerce

The backdoor code is the **More_egg** JavaScript backdoor malware ... and Metasploit **and PowerShell** to move laterally and deploy malware.

4 days ago



2019

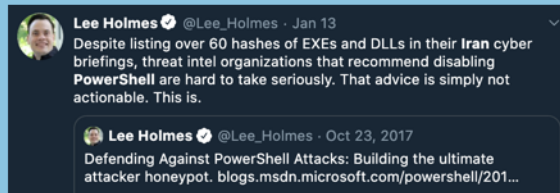
Script kiddiez are still widely using PowerShell & Empire

<https://twitter.com/xorrior/status/1156626181107736576>



2019

Enterprises and vendors struggle to identify malicious PowerShell



	<p>Mitigation</p> <ul style="list-style-type: none"> Configure Office security settings enable Protected View, to execute within a sandbox environment, and to block macros through Group Policy. Other types of virtualization and application microsegmentation may also mitigate the impact of compromise. Turn off unused features or restrict access to scripting engines such as VBScript or scriptable administration frameworks such as PowerShell.
Scripting	<p>Detection</p> <ul style="list-style-type: none"> Examine scripting user restrictions. Evaluate any attempts to enable scripts running on a system that would be considered suspicious.
PowerShell	<p>Mitigation</p> <ul style="list-style-type: none"> Set PowerShell execution policy to execute only signed scripts. Remove PowerShell from systems when not needed, but a review should be performed to assess the impact to an environment, since it could be in use for many legitimate purposes and administrative functions. Disable/restrict the WinRM Service to help prevent uses of PowerShell for remote execution. Restrict PowerShell execution policy to administrators. <p>Detection</p> <ul style="list-style-type: none"> If PowerShell is not used in an environment, looking for PowerShell execution may detect malicious activity. Monitor for loading and/or execution of artifacts associated with PowerShell specific assemblies, such as System.Management.Automation.dll (especially to unusual process names/locations). Turn on PowerShell logging to gain increased fidelity in what occurs during execution (which is applied to .NET invocations).

2020

Iran's cyberattacks included commodity PowerShell malware.
US CERT guidance was both helpful and not helpful.

https://twitter.com/Lee_Holmes/status/1216703900792655875

Iran CERT Bulletin Calling Out PowerShell

<https://www.us-cert.gov/ncas/alerts/aa20-006a>

PowerShell Editions

Windows PowerShell

Built-in on Windows since 7/08R2

Upgrade with Windows Mgmt Framework

Version 5.1 is “complete”

PowerShell Core

Open beta 2016 / Released 2018

Cross-platform: Win/Lin/Mac/ARM

Legit shell: pwsh

Same policies as Windows

All new development going here

Open source: <https://github.com/PowerShell/PowerShell>

Open community call 3rd Thursdays: <https://aka.ms/pscommunitycall>

```
MACBOOKPRO:~ goatee$ pwsh
PowerShell 7.0.2
Copyright (c) Microsoft Corporation. All rights reserved.

https://aka.ms/powershell
Type 'help' to get help.

PS /Users/goatee> $PSVersionTable

Name                           Value
----                           -
PSVersion                      7.0.2
PSEdition                      Core
GitCommitId                    7.0.2
OS                             Darwin 19.5.0 Darwin Kernel_
Platform                      Unix
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0}
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1
WSManStackVersion              3.0

PS /Users/goatee> Get-Variable Is*

Name                           Value
----                           -
IsCoreCLR                     True
IsLinux                       False
IsMacOS                      True
IsWindows                    False
```

<https://devblogs.microsoft.com/powershell/getting-started-with-powershell-core-on-windows-mac-and-linux/>

<https://aka.ms/pscommunitycall>

<https://github.com/PowerShell/PowerShell>

How to PowerShell Security

PowerShell Hygiene

Level 0: Upgrade old machines to Windows PowerShell 5.1 (WMF 5.1)

Disable PowerShell Version 2 feature

Level 1: Implement and monitor: logging & transcription

Level 2: Enable and secure remoting (default endpoint, firewall)

Level 3: Block evil code: AppLocker / DeviceGuard / AMSI

Feature Matrix By Operating System

Built-in but not enabled or configured by default

Upgrade your schtuff

	Win/Mac/Linux	Windows 10	Windows 8.1	Windows 8	Windows 7
	PS Core	WS 2016/19	WS 2012 R2	WS 2012	WS 2008 R2
Module Logging	Built-In	Built-In	Built-In	Built-In	WMF 5.1
Script Block Logging			WMF 5.1	WMF 5.1	
Transcription			Available	Available	
PSReadline Command History			Available	Available	
Anti-Malware Scan Interface	n/a		n/a	n/a	n/a

WMF = Windows Management Framework (PowerShell upgrade)

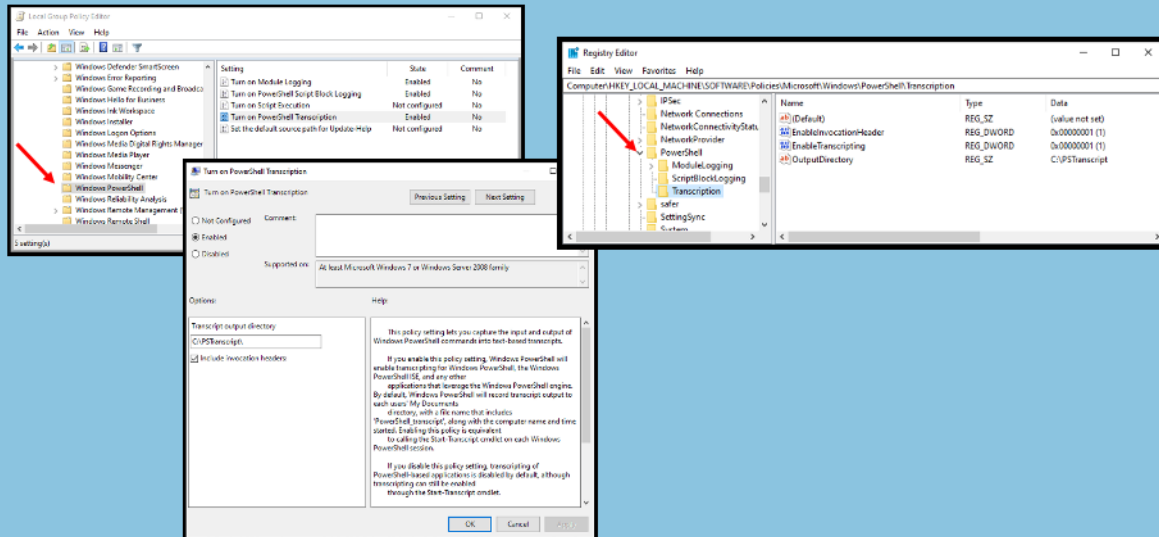
Events & Log Files

	Event Log	Event IDs
Module Logging	Windows PowerShell	400, 800, 403
Module Logging	Microsoft-Windows-PowerShell/Operational & PowerShellCore/Operational	4103
Script Block Logging	Microsoft-Windows-PowerShell/Operational & PowerShellCore/Operational	4105, 4104, 4106
Anti-Malware Scan Interface	Microsoft-Windows-Windows Defender/Operational	1116, 1117
Transcription ♦	HKLM:\Software\Policies\Microsoft\Windows\PowerShell\Transcription\OutputDirectory HKLM:\Software\Policies\Microsoft\PowerShellCore\Transcription\OutputDirectory	
PSReadline Command History ♦♦	C:\Users*\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline*.txt (Get-PSReadlineOption).HistorySavePath	

♦ Path configured by policy

♦♦ Default path recommended not to change

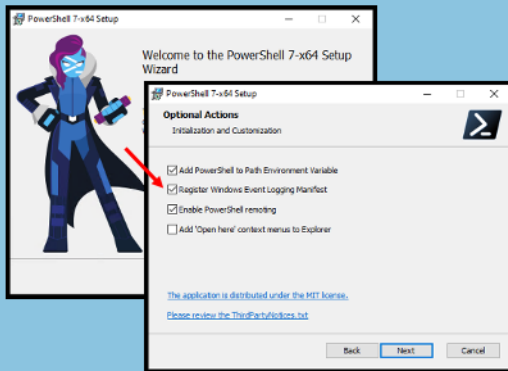
Policies on Windows – Windows PowerShell



https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_group_policy_settings?view=powershell-7.1

PowerShell Core Install Warning

PowerShell Core 7.x installer requires you to register the event log manifest for logging to work.



```
msiexec.exe /package PowerShell-7.0.1-win-x64.msi /quiet  
ADD_EXPLORER_CONTEXT_MENU_OPENPOWERSHELL=1  
ENABLE_PSREMOTING=1 REGISTER_MANIFEST=1
```

```
PS C:\Program Files\PowerShell\7> dir *instrum*,*.ps1  
  
Directory: C:\Program Files\PowerShell\7  
  
Mode                LastWriteTime         Length Name  
----                -  
-a---             2/12/2020  7:12 PM        101456 PowerShell.Core.Instrumentation.dll  
-a---             6/9/2020   9:59 PM        220663 PowerShell.Core.Instrumentation.man  
-a---             6/9/2020  10:23 PM         21919 Install-PowerShellRemoting.ps1  
-a---             6/9/2020  10:23 PM         15869 InstallPSCorePolicyDefinitions.ps1  
-a---             6/9/2020  10:23 PM         16162 RegisterManifest.ps1  
  
PS C:\Program Files\PowerShell\7> wevtutil ep | select-string "powershell"  
Microsoft-Windows-PowerShell  
Microsoft-Windows-PowerShell-DesiredStateConfiguration-FileDownloadManager  
PowerShell  
PowerShellCore  
  
PS C:\Program Files\PowerShell\7>
```

<https://docs.microsoft.com/en-us/powershell/scripting/install/installing-powershell-core-on-windows?view=powershell-7#installing-the-msi-package>

Policies on Windows – PowerShell Core

The image is a composite of three screenshots illustrating PowerShell Core policies on Windows.

Top Left: PowerShell Command Window

```
PS C:\Program Files\PowerShell\7> dir *core*policy*

Directory: C:\Program Files\PowerShell\7

Mode                LastWriteTime         Length Name
----                -
-a---             6/9/2020 10:23 PM        15869 InstallPSCorePolicyDefinitions.ps1
-a---             6/9/2020  9:59 PM         9675 PowerShellCoreExecutionPolicy.adml
-a---             6/9/2020  9:59 PM         6201 PowerShellCoreExecutionPolicy.admx

PS C:\Program Files\PowerShell\7> .\InstallPSCorePolicyDefinitions.ps1
```

Top Right: Local Group Policy Editor

The Local Group Policy Editor window shows the 'PowerShell Core' section under 'Computer Configuration' > 'Administrative Templates'. The 'Turn on PowerShell Transcription' policy is highlighted with a red arrow. The 'Settings' pane on the right shows the policy is 'Not configured'.

Bottom Left: Registry Editor

The Registry Editor window shows the path 'Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\PowerShellCore\Transcription'. The 'Transcription' folder is highlighted with a red arrow. The right pane shows the following registry values:

Name	Type	Data
(Default)	REG_SZ	(value not set)
EnableInvocationHeader	REG_DWORD	0x00000001 (1)
EnableTranscription	REG_DWORD	0x00000001 (1)
OutputDirectory	REG_SZ	C:\PS\TranscriptCore\
UseWindowsPowerShellPolicySetting	REG_DWORD	0x00000001 (1)

Bottom Right: Turn on PowerShell Transcription dialog box

The dialog box shows the 'Turn on PowerShell Transcription' policy. The 'Options' section has the following settings:

- ☒ Use Windows PowerShell Policy setting.
- ☒ C:\PS\TranscriptCore\
- ☒ Include invocation headers.

The 'Help' section explains that this policy setting lets you capture the input and output of PowerShell Core commands into text-based transcripts. It also notes that enabling this policy is required to call the Start-Transcript cmdlet on each PowerShell Core session.

https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_group_policy_settings?view=powershell-7.1

Policies on Windows/MacOS/Linux – PowerShell Core

```
PS C:\Program Files\PowerShell\7> ls *.json

Directory: C:\Program Files\PowerShell\7

Mode                LastWriteTime         Length Name
----                -
-a---             6/9/2020 10:02 PM           406 powershell.config.json
-a---             6/9/2020 10:01 PM       134189 pwsh.deps.json
-a---             6/9/2020 10:01 PM          526 pwsh.runtimeconfig.json
```

```
PS /Users/goatee> cd $PSHOME
PS /usr/local/microsoft/powershell/7> ls -l *.json
-rw-r--r--  1 root  wheel   591 Jul 15 12:18 powershell.config.json
-rw-r--r--  1 root  wheel 113474 Jun  9 18:01 pwsh.deps.json
-rw-r--r--  1 root  wheel  414 Jun  9 18:01 pwsh.runtimeconfig.json
```

```
{
  "Microsoft.PowerShell:ExecutionPolicy": "RemoteSigned",
  "PowerShellPolicies": {
    "ScriptBlockLogging": {
      "EnableScriptBlockInvocationLogging": false,
      "EnableScriptBlockLogging": true
    },
    "ModuleLogging": {
      "EnableModuleLogging": true,
      "ModuleNames": [
        "*"
      ]
    },
    "Transcription": {
      "EnableTranscripting": true,
      "EnableInvocationHeader": true,
      "OutputDirectory": "/var/tmp/pstranscripts/"
    }
  },
  "LogLevel": "verbose"
}
```

https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_logging_non-windows?view=powershell-7.1#configuring-logging-on-a-non-windows-system

The file powershell.config.json is a JSON formatted file residing in the PowerShell \$PSHOME directory.

```
{
  "Microsoft.PowerShell:ExecutionPolicy": "RemoteSigned",
  "PowerShellPolicies": {
    "ScriptBlockLogging": {
      "EnableScriptBlockInvocationLogging": false,
      "EnableScriptBlockLogging": true
    },
    "ModuleLogging": {
      "EnableModuleLogging": true,
      "ModuleNames": [
        "*"
      ]
    },
    "Transcription": {
      "EnableTranscripting": true,
      "EnableInvocationHeader": true,
      "OutputDirectory": "/var/tmp/pstranscripts/"
    }
  },
  "LogLevel": "verbose"
}
```

```
"Transcription": {  
  "EnableTranscripting": true,  
  "EnableInvocationHeader": true,  
  "OutputDirectory": "/var/tmp/pstranscripts/"  
},  
"LogLevel": "verbose"  
}
```



Pro Tip: PowerShell policy changes will take effect in a new session.

DEMO

Be careful out there...

Transcription known issues:

- Citrix can crash

- Active Directory Administrative Center can hang

- Microsoft SCOM management script failures (may automatically be disabled in 2016+)

VSCode PowerShell plugin may spam the logs with Script Analyzer activity.

Transcription hardening on PSv4 will crash PowerShell.

Event log hardening may require a reboot to take full effect.

What issues have you encountered?

Resources

PSPolicy Module – filling the gaps after GPO

Set the policies

Set event log size

Harden the transcripts and logs

Clean the transcript files

Search all logging locations for a string - *foot gun warning*

Work in progress

Only supports Windows PowerShell 5.1 so far

<https://github.com/GoateePFE/PSPolicy>

Free Resources

Hands-on-lab you can do at home (packed with practical details):

<https://github.com/GoateePFE/PowerShellSummit2019>

Instructions for installing PowerShell on any OS:

<https://docs.microsoft.com/en-us/powershell/scripting/install/installing-powershell>

PDF of this deck:

<https://github.com/GoateePFE/BSidesCBUS2020>

Contact me with questions or feedback:

ashley.mcglone@tanium.com Twitter: @GoateePFE

Tanium Can Help



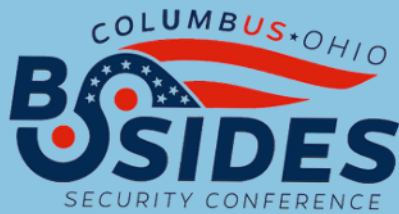
This presentation is a public education thing, NOT a vendor commercial.

That being said, Tanium can help automate these policies and data collection at scale.

<https://bit.ly/TaniumPS>

Contact me if you would like more information:

ashley.mcglone@tanium.com Twitter: @GoateePFE



PowerShell pew pew pew:
skilz 4 blue team

Mr. Ashley McGlone

Tanium, TAM
ashley.mcglone@tanium.com
@GoateePFE

In the words of Mark Minasi, “Use your powers for good and not for evil.”