Defense Against the Lateral Arts: Detecting and Preventing Impacket's Wmiexec

August 31, 2022 | Stephan Wolfert | From The Front Lines

- Impacket, an open source collection of Python modules for manipulating network protocols, contains several tools for remote service execution, Windows credential dumping, packet sniffing and Kerberos manipulation

Featured
Recent
Video
Category
Start Free Trial

Impacket's wmiexec.py ("wmiexec") is a popular tool used by red teams and threat actors alike. The CrowdStrike Services team commonly sees threat actors leveraging wmiexec

executable on Windows systems, remotely executing data exfiltration tools such as Rclone, and Cobalt Strike beacons for lateral movement and command-and-control operations.

Impacket's suite of tools is extremely versatile and is low impact, making detection more difficult compared to other threat actor tool sets. This blog deep dives into wmiexec usage seen from multiple incident response investigations, and describes indicators to help defenders detect wmiexec.

Wmiexec Overview

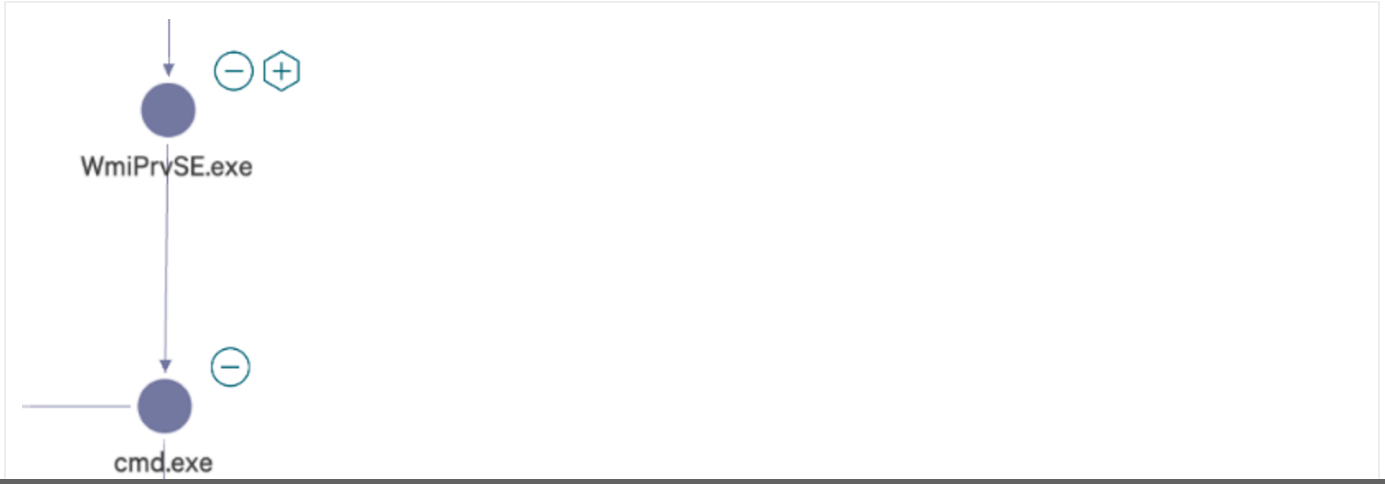
Wmiexec relies on the Windows native service known as Windows Management Instrumentation (WMI). Microsoft defines WMI as "the infrastructure for management data and operations on Windows-based operating systems." While WMI has legitimate use-cases, threat actors commonly use WMI to move laterally.

Wmiexec allows a threat actor to execute commands on a remote system and/or establish a semi-interactive shell on a remote host. The remote connection and command

execution of wmiexec.py will establish their connection with DCOM/RPC on port 135, the response back to the threat actor system is sent via the Server Message Block ("SMB")

Initial Indicators

When hunting for wmiexec, defenders should look for WMI usage. A defender's first step should be to analyze the process relationship involving a parent process known as `WMIPRVSE.EXE`. Suspicious processes such as `CMD.EXE` or `POWERSHELL.EXE` running as a child process to `WMIPRVSE.EXE` are a red flag. Most commonly, and by default, wmiexec will use a child process of `CMD.EXE`. A common indicator of wmiexec is the command line switches of the `CMD.EXE` process, which is somewhat unique. An example of executing a tasklist using wmiexec would establish a process relationship similar to the image in Figure 1. Throughout this blog we will often refer to the publicly available source code on Impacket's GitHub repository.



Featured
Recent
Video
Category
Start Free Trial

Figure 1. Parent process relationship

Understanding *wmiexec* Command Execution

checking the command is run silently. Secondly, the parameter `/c`

is set to stop after the command specified by the string is carried out.

```
127                 self.__shell = 'cmd.exe /Q /c '
```

Figure 2. Code example calling of cmd.exe

Identifying commands issued with the default `cmd.exe /Q /c` arguments are an indicator that wmiexec may be in use but note that these are parameters which can be used for legitimate purposes. To further validate the identification of wmiexec usage, another indicator is command redirection. During execution of *wmiexec*, the command is redirected to a file created on the remote host's local ADMIN$ share by default. The ADMIN$ share aligns with the file path `C:\Windows\`.

```
295                 command += ' 1> ' + '\\\\127.0.0.1\\%s' % self.__share + self.__output + ' 2>&1'
```

Figure 3. Code example of redirected output to a file (Click to enlarge)

As shown in Figure 3, on line 295 of the wmiexec code, the command variable has a few variables that are appended with additional data, concatenating the `/Q /c` switches with the command being run and the redirection. While this full command line is a great indicator of wmiexec usage, the variable `__output` (shown in Figure 3 as `self.__output`) is the name of the temporary file written to disk and creates additional forensic artifacts on disk. When the tool concatenates the command parameters

Featured
Recent
Video
Category
Start Free Trial

command is given a filename of two underscores, `__`, followed by the current time in EPOCH. This is important for two reasons: the presence of this file indicates execution of (or attempted execution of) wmiexec and also it will give us a time of execution. As previously mentioned, this file will reside in `C:\Windows\`, which is remotely accessible

actor, which can also be useful to defenders (more on this later).

```
46   OUTPUT_FILENAME = '__' + str(time.time())
```

```
125              self.__output = '\\' + OUTPUT_FILENAME
```

Figure 5. Code example of the output filename

```
69 __1645636159.6177979
```

Figure 6. Example of output file written to disk

Cleanup Operations

The output file is not always present on disk because wmiexec, upon successful and complete execution, will clean up after itself. Most commonly this file is left behind for one of two reasons: a threat actor prematurely canceled a command before it has completed (i.e., CTRL+C), or wmiexec failed or was prevented from executing. In either scenario, if the wmiexec script is unable to complete execution it will not reach line 285, shown in Figure 7, to clean up the file on disk. Also of note, incomplete execution of wmiexec does not necessarily mean the threat actor was unable to successfully execute their command.

Featured
Recent
Video
Category
Start Free Trial

established that appears interactive, but each command executed will be sent through the wmiexec channel and include the standard process execution `cmd.exe /Q /C <command> 1> \\\\127.0.0.1\ADMIN$\__<EPOCHTIME> 2>&1` command format. In

since nothing was supplied to nslookup, and although this was done in a test environment, CrowdStrike Services has seen this exact occurrence happen during real incident response engagements.

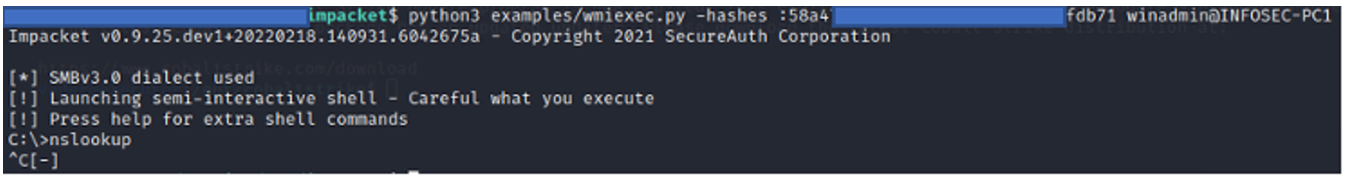


Figure 8. Execution of wmiexec.py to interrupt nslookup execution (Click to enlarge)

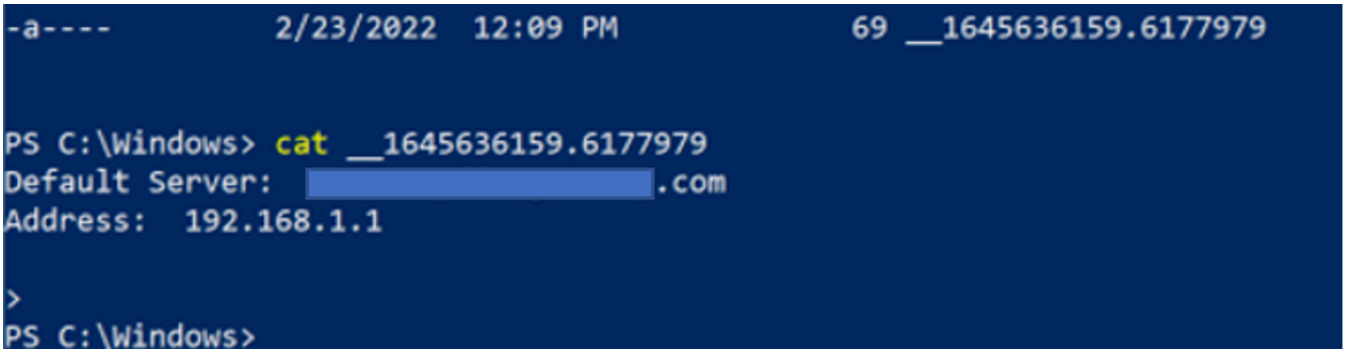


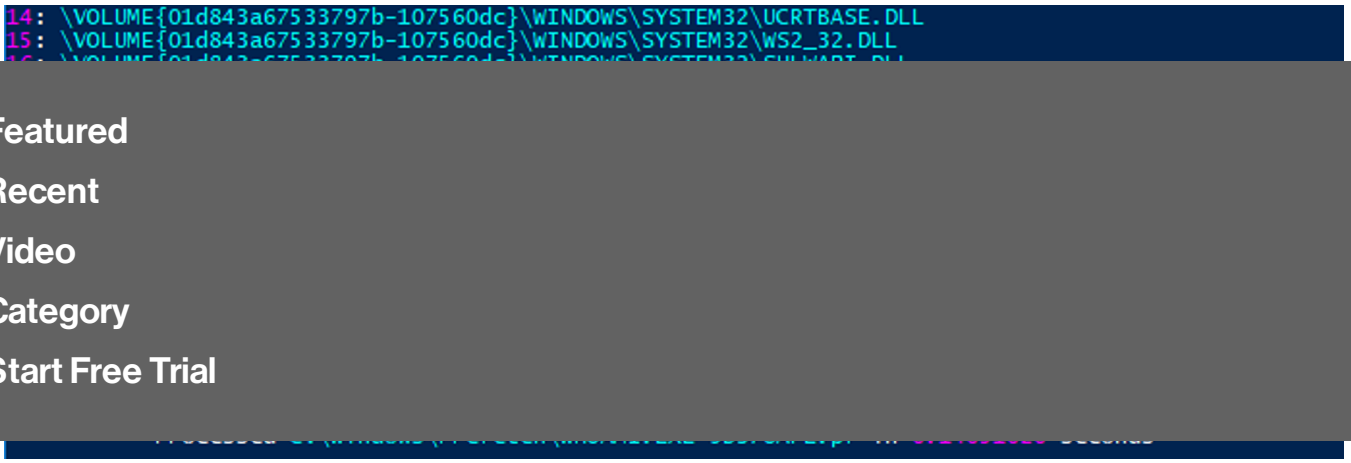
Figure 9. Examining the leftover file contents

Another example of failure to clean up and artifacts left on disk is shown in Figures 10 and 11. In Figure 10, wmiexec is executed similarly as before, but this time `powershell.exe` is executed. Since running the command powershell.exe would open a new PowerShell prompt, issuing another abort command (CTRL+C) will result in the output file not being cleaned up. Figure 11 shows the header of a new PowerShell window in the contents of

Figure 11. Examining the leftover file contents of PowerShell output (Click to enlarge)

The temporary files discussed here can be a valuable indicator of wmiexec execution even when they are successfully cleaned up. Windows Prefetch files are used by the Microsoft Windows operating system to improve application start-up performance. Prefetch is a common forensic artifact located in `C:\Windows\Prefetch` that can be used to identify process execution along with contextual information related to the file that was executed.

As shown in Figure 12, contextual information can be parsed from Prefetch using a tool such as PECmd. This contextual information includes Dynamic Link Libraries (DLLs) and other files used by the process that was executed. In this example, a variety of DLLs are referenced in the Prefetch file for `whoami.exe` as well as the temporary files associated with wmiexec. The temporary files previously discussed cannot be extracted from the Prefetch file — but, this example does show that Prefetch data contains evidence that `whoami.exe` was executed with wmiexec.

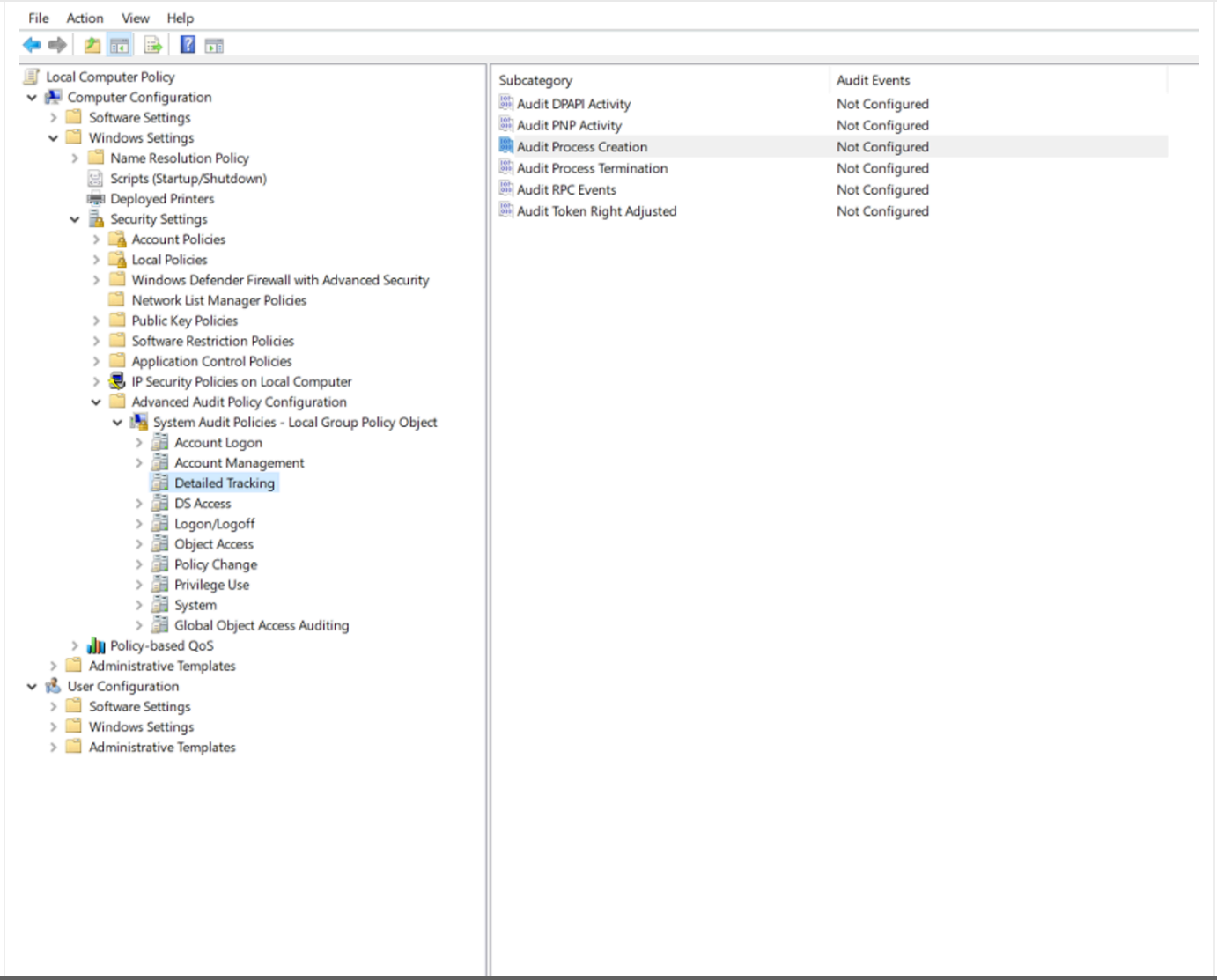


Featured
Recent
Video
Category
Start Free Trial

Figure 12. Examining parsed Prefetch data to identify a relation between whoami and temporary wmiexec files

Key visibility into potential wmiexec execution comes from command-line process execution. Process execution will give a defender more definitive evidence and visibility

data on the command line execution with the right settings enabled. Unfortunately, Security Event ID 4688 is **not** enabled by default. After enabling Event ID 4688, defenders must also configure it for full command line auditing, which is described below. Note: When introducing any recommendations for security or visibility enhancements to your environment, proper testing should be conducted to ensure it will not affect business operations negatively. To enable Event ID 4688, open the Local Group Policy Editor and do the following (shown in Figure 13): Local Group Policy Editor > Open Computer Configuration > Open Windows Settings > Security Settings > Advanced Audit Policy Configuration > System Audit Policies > Detailed Tracking > Enabled Audit Process



Featured
Recent
Video
Category
Start Free Trial

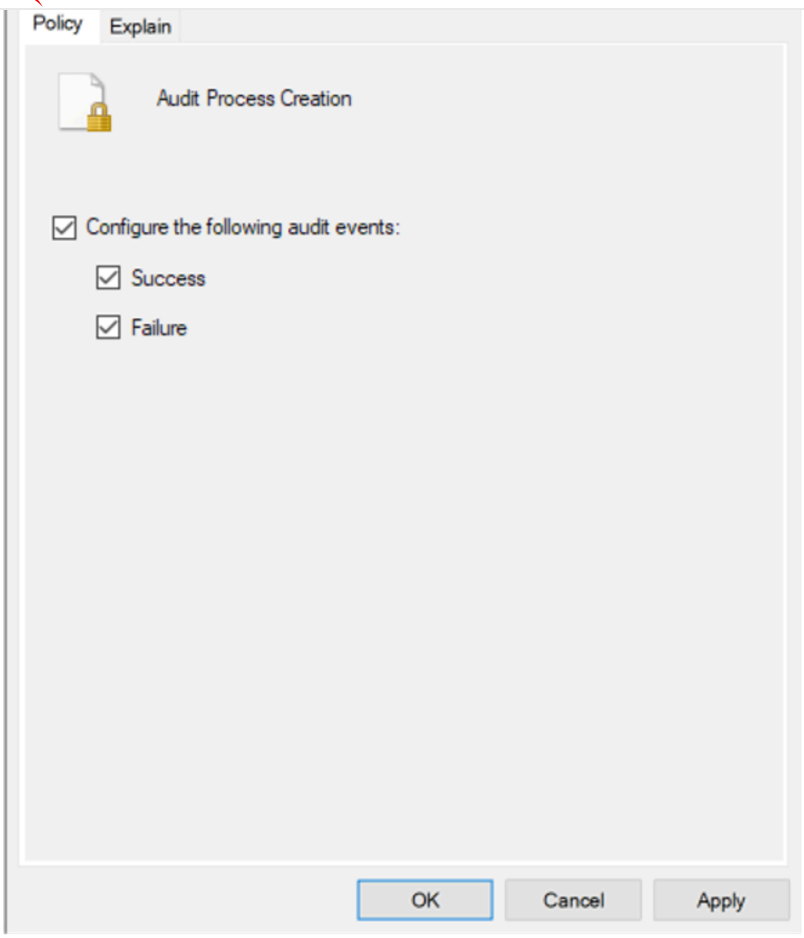
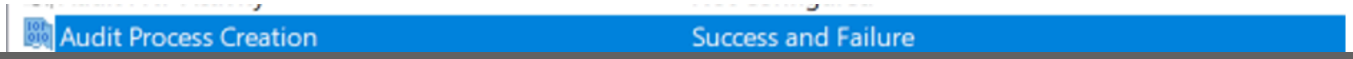


Figure 14. Enabling Success and Failure events

Finally, check to make sure auditing is enabled, as shown in Figure 15.



Featured
Recent
Video
Category
Start Free Trial

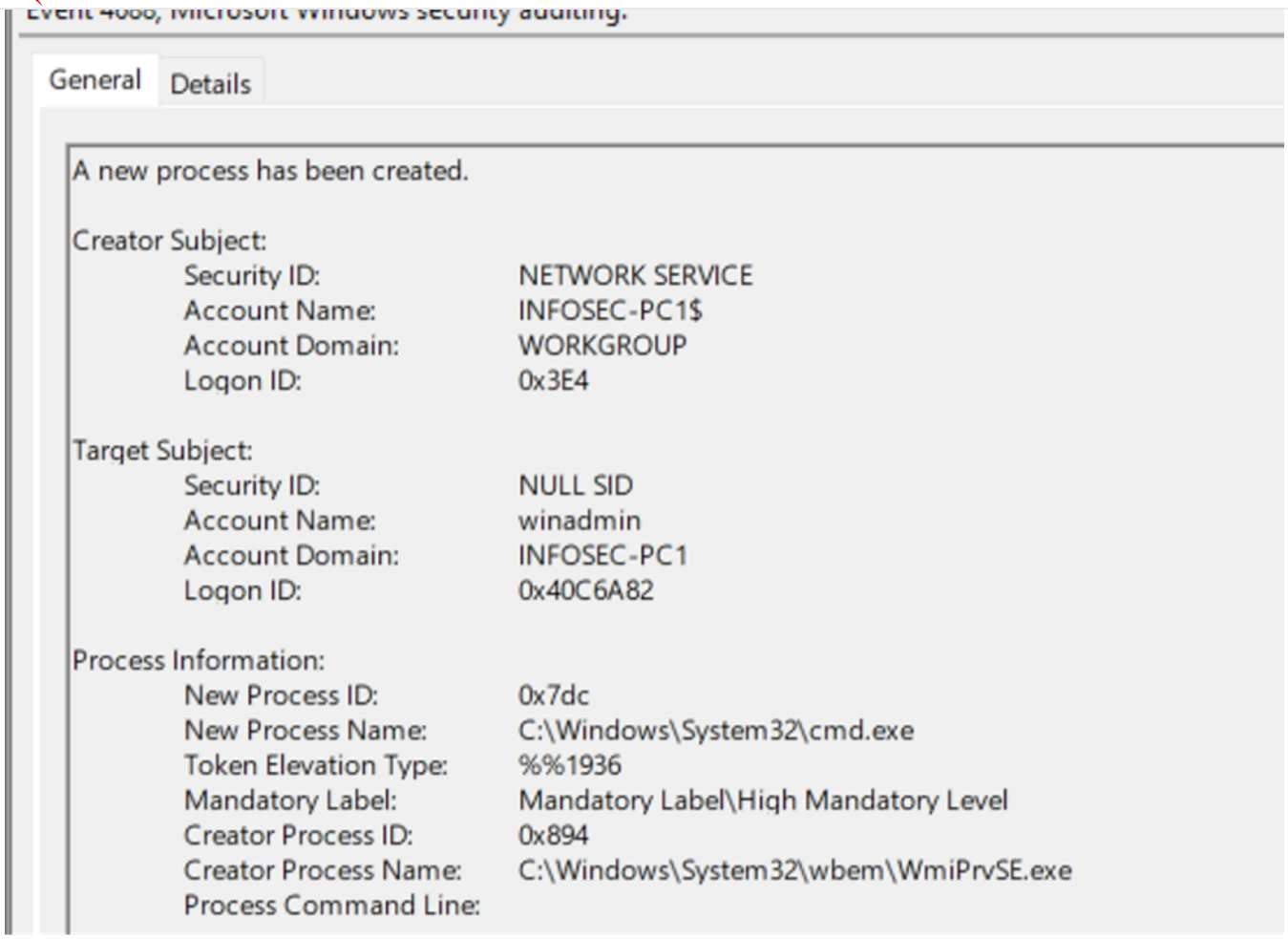


Figure 16. Event ID 4688 enabled showing process creation of wmiexec

Although visibility into parent/child processes on a system helps defenders, the full process command line in Figure 16 is blank because it must be enabled separately. In

Featured
Recent
Video
Category
Start Free Trial

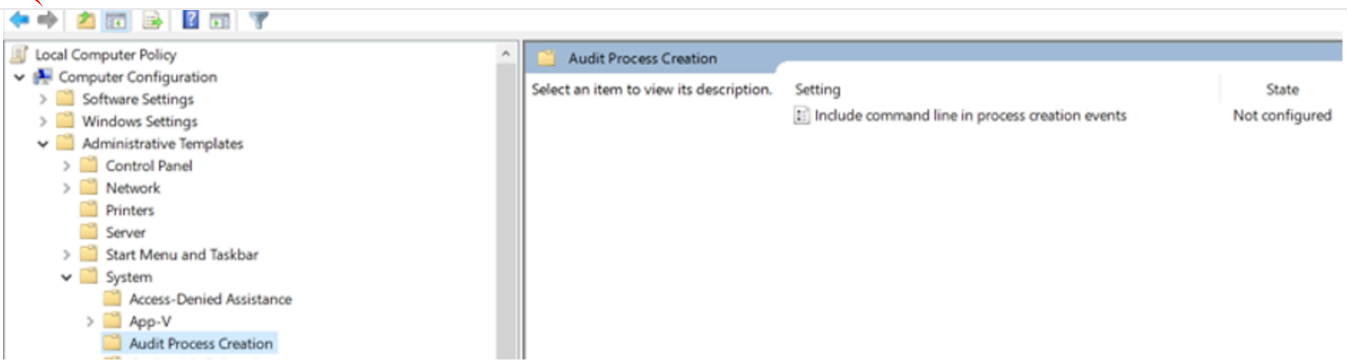
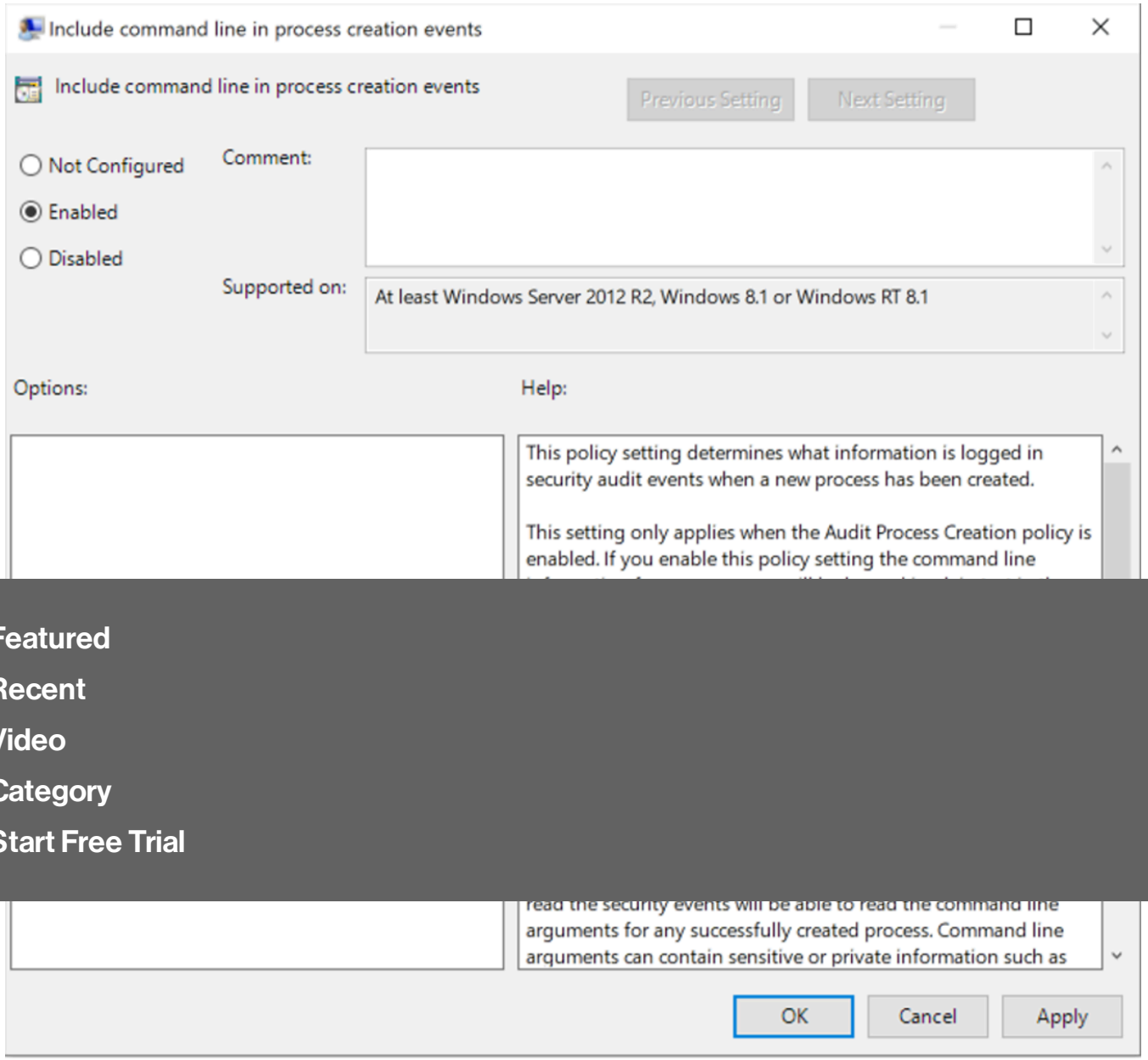
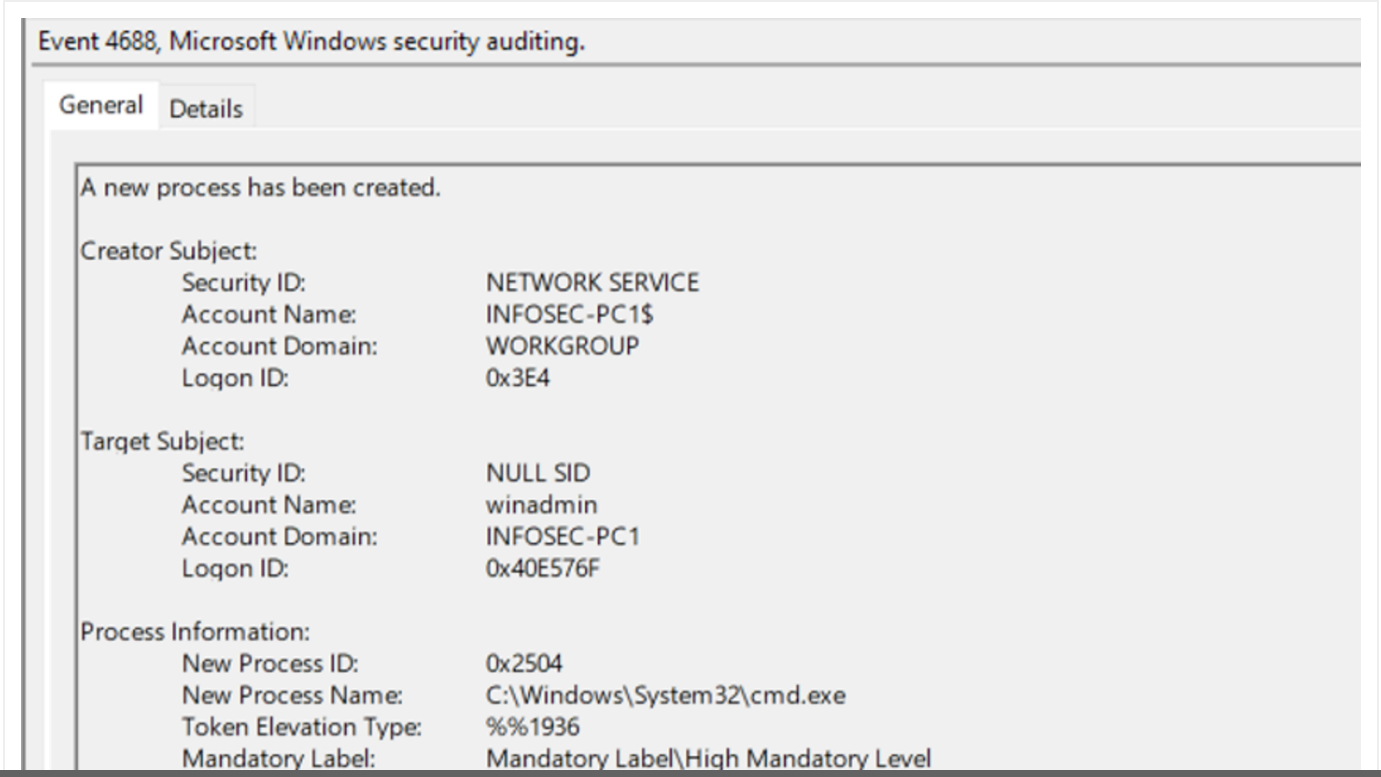


Figure 17. Enabling process command line for Event ID 4688



Featured
Recent
Video
Category
Start Free Trial

the exact commands executed is recorded, rather than just parent/child process execution. Forwarding this data to a centralized security information and event management (SIEM) tool can allow defenders to set up alerts and dashboards to track process executions. As shown in Figure 19, Event ID 4688 is now logging the entire command line, giving a defender the ability to see wmiexec was used to remotely execute the command `hostname`.



Prevention and Mitigation

interactive shell will give the threat actor the ability to run commands and while visibility into these attacks is critical, mitigating them is the ultimate goal.

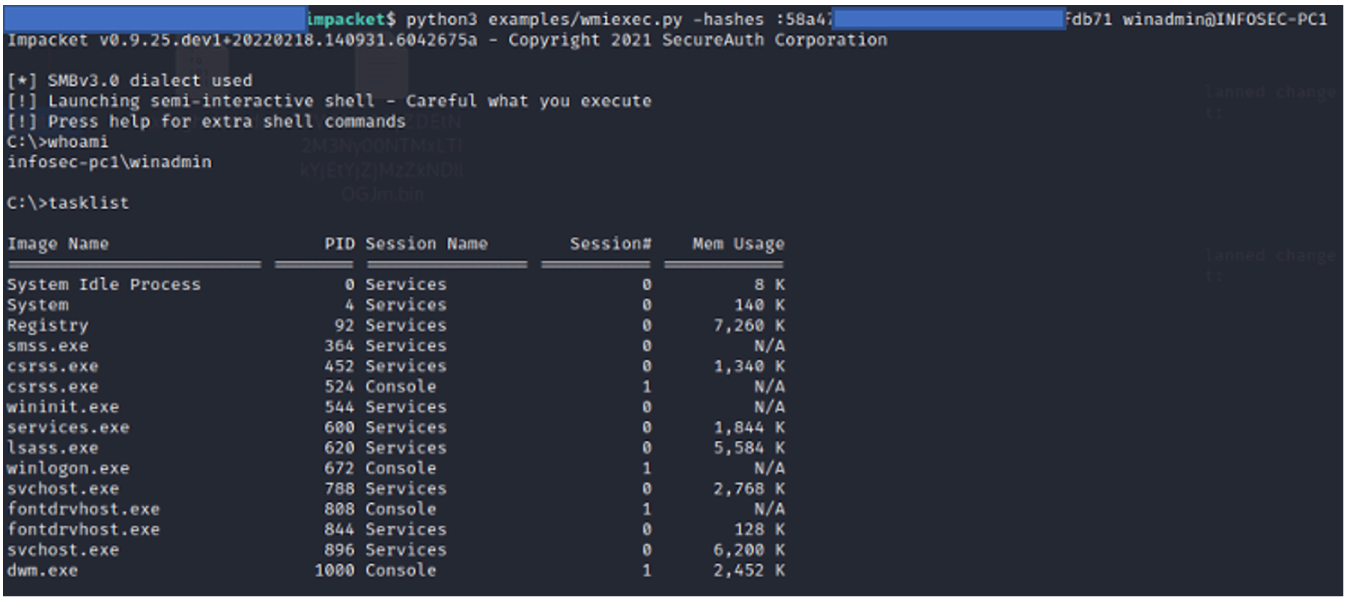


Figure 20. Threat actor execution of remote commands with wmiexec (Click to enlarge)

CrowdStrike Falcon® Insight™ endpoint detection and response will collect granular data on process execution in real time, going into more detail than the standard process creation logging available in Windows. Indicators of attack (IOAs) are a method used to create detections and preventions for wmiexec. Commonly on CrowdStrike Services Red Team/Blue Team assessments, CrowdStrike experts will assist internal teams to configure an IOA that will allow detection and prevention of wmiexec. In Figure 21, having a

Featured
Recent
Video
Category
Start Free Trial

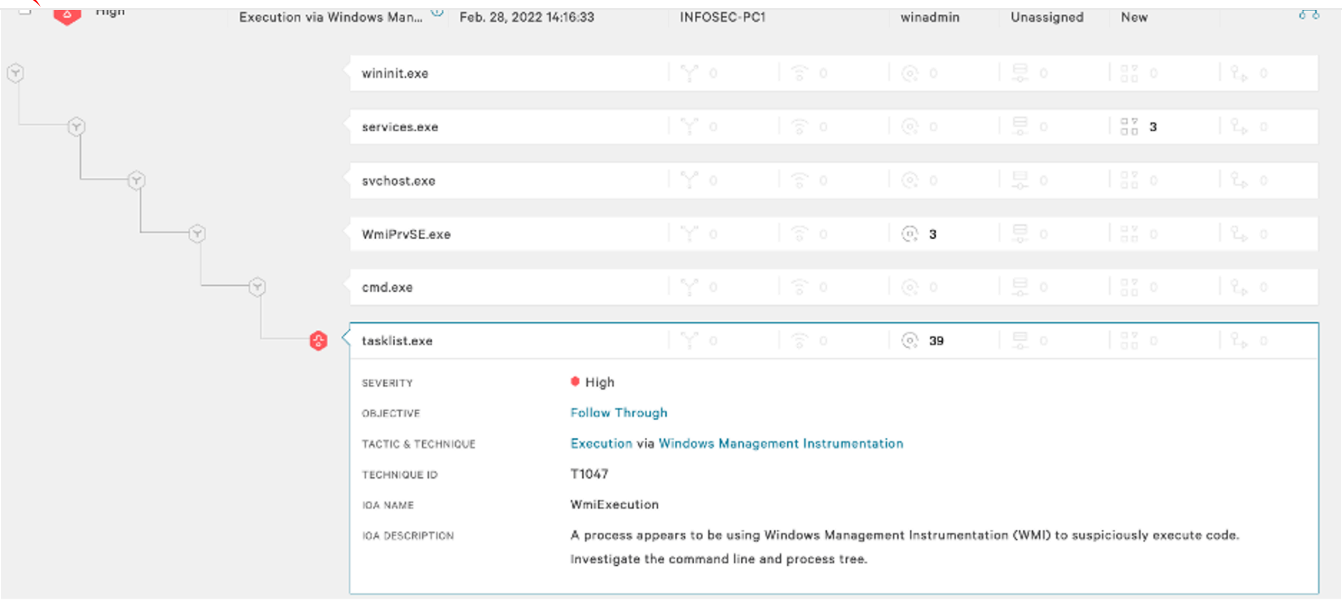


Figure 21. CrowdStrike Falcon® IOA for wmiexec (Click to enlarge)

Each of the processes that spawned the suspected malicious process can be examined to identify a gold mine of information, including the host where the request originated and full command line parameters, as shown in Figure 22.

Figure 22. Process command line for wmiexec running tasklist

The prevention will restrict the threat actor's ability to issue remote commands to systems that have an active CrowdStrike Falcon® endpoint protection agent. As shown in

Featured
Recent
Video
Category
Start Free Trial

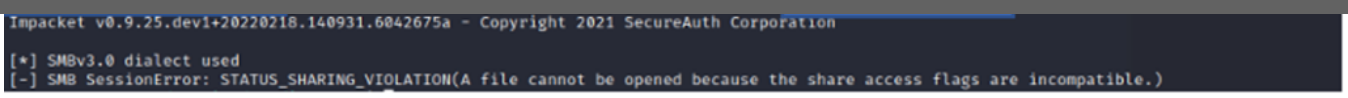


Figure 24. Failure to execute wmiexec tasklist command due to Falcon Insight IOA (Click to enlarge)

Falcon Insight provides detection data for events with the appropriate prevention and

process, `cmd.exe`; the remote host that executed the command, `192.168.1.211` (threat actor Testing System); and the full process command line. This is all captured in real time and enables defenders to investigate and respond quickly. The process command line should look familiar and shows execution of a handful of commands (`cd`, `whoami` and `tasklist`).

Figure 25. Event search example of wmiexec commands issued (Click to enlarge)

Falcon Forensics

When looking historically at the artifacts discussed in this blog, Falcon Forensics can be a great asset to a defender's investigation. Falcon Forensics is a run-once script that is easily deployed through RTR or other deployment tools that will capture a variety of forensic artifacts used for forensic triage of a system. The most beneficial advantage to using Falcon Forensics is the variety of sourcetypes and triage data matched with the ability to perform investigations at scale across an entire environment. Considering the artifacts discussed in this blog, there are specific examples that can be used to find

artifacts. Using Falcon Forensics across your environment can look for these artifacts at

prefetch	\VOLUME{01d843a67533797b-107560dc}\WINDOWS\SYSTEM32\IMM32.DLL	C:\Windows\Prefetch\WHOAMI.EXE-9D378AFE.pf
prefetch	\VOLUME{01d843a67533797b-107560dc}\WINDOWS__1658938223.9829855	C:\Windows\Prefetch\WHOAMI.EXE-9D378AFE.pf
prefetch	\VOLUME{01d843a67533797b-107560dc}\WINDOWS__1658938368.1822846	C:\Windows\Prefetch\WHOAMI.EXE-9D378AFE.pf
prefetch	\VOLUME{01d843a67533797b-107560dc}\WINDOWS__1658948944.656154	C:\Windows\Prefetch\WHOAMI.EXE-9D378AFE.pf
prefetch	\VOLUME{01d843a67533797b-107560dc}\WINDOWS\GLOBALIZATION\SORTING\SORTDEFAULT.NLS	C:\Windows\Prefetch\WHOAMI.EXE-9D378AFE.pf
prefetch	\VOLUME{01d843a67533797b-107560dc}\WINDOWS\SYSTEM32\EN-US\WHOAMI.EXE.MUI	C:\Windows\Prefetch\WHOAMI.EXE-9D378AFE.pf

Figure 26. Falcon Forensics event search example of wmiexec artifacts in Prefetch (Click to enlarge)

In addition to the Prefetch sourcetype, pslist can be used to identify the previously discussed command line artifacts associated with wmiexec that may be still running at the time of Falcon Forensics deployment. Shown in Figure 27, since PowerShell and nslookup will both hang on execution, the processes were still running, which allowed for visibility into the process execution.

sourcetype	cmdline	name
pslist	cmd.exe /Q /c powershell.exe 1> \\127.0.0.1\ADMIN$__1658948979.8670597 2>&1	cmd.exe
pslist	cmd.exe /Q /c nslookup 1> \\127.0.0.1\ADMIN$__1658948959.0443158 2>&1	cmd.exe

Figure 27. Falcon Forensics event search example of wmiexec artifacts in pslist (Click to enlarge)

Lastly, files written to disk are a strong indicator of threat actor activity. In Figure 27 an example of wmiexec execution that failed to clean up is easily identifiable using Falcon Forensics to search for the recognizable common file name format, `__<EPOCHTIME>`, in `C:\Windows`.

sourcetype	name	path
dirlist	__1658948979.8670597	C:\Windows\
dirlist	__1658948959.0443158	C:\Windows\

Figure 28. Falcon Forensics event search example of wmiexec artifacts in dirlist (Click to enlarge)

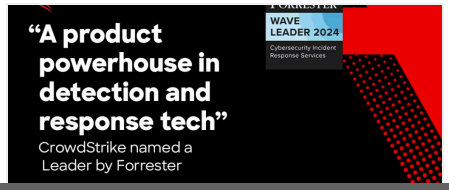
Featured
Recent
Video
Category
Start Free Trial

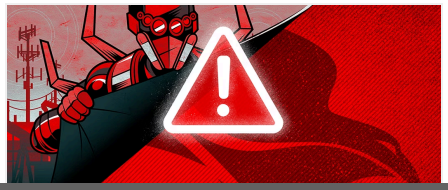
sets to identify avenues for detection and prevention.

Additional Resources

- *Read about adversaries tracked by CrowdStrike in 2021 in the 2022 CrowdStrike Global Threat Report.*
- *Learn more about the CrowdStrike Falcon® platform by visiting the product webpage.*
- *Test CrowdStrike next-gen AV for yourself. Start your free trial of Falcon Prevent™ today.*

Related Content



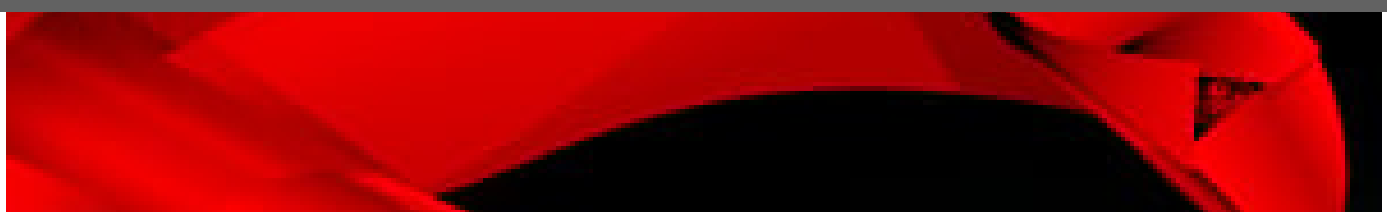


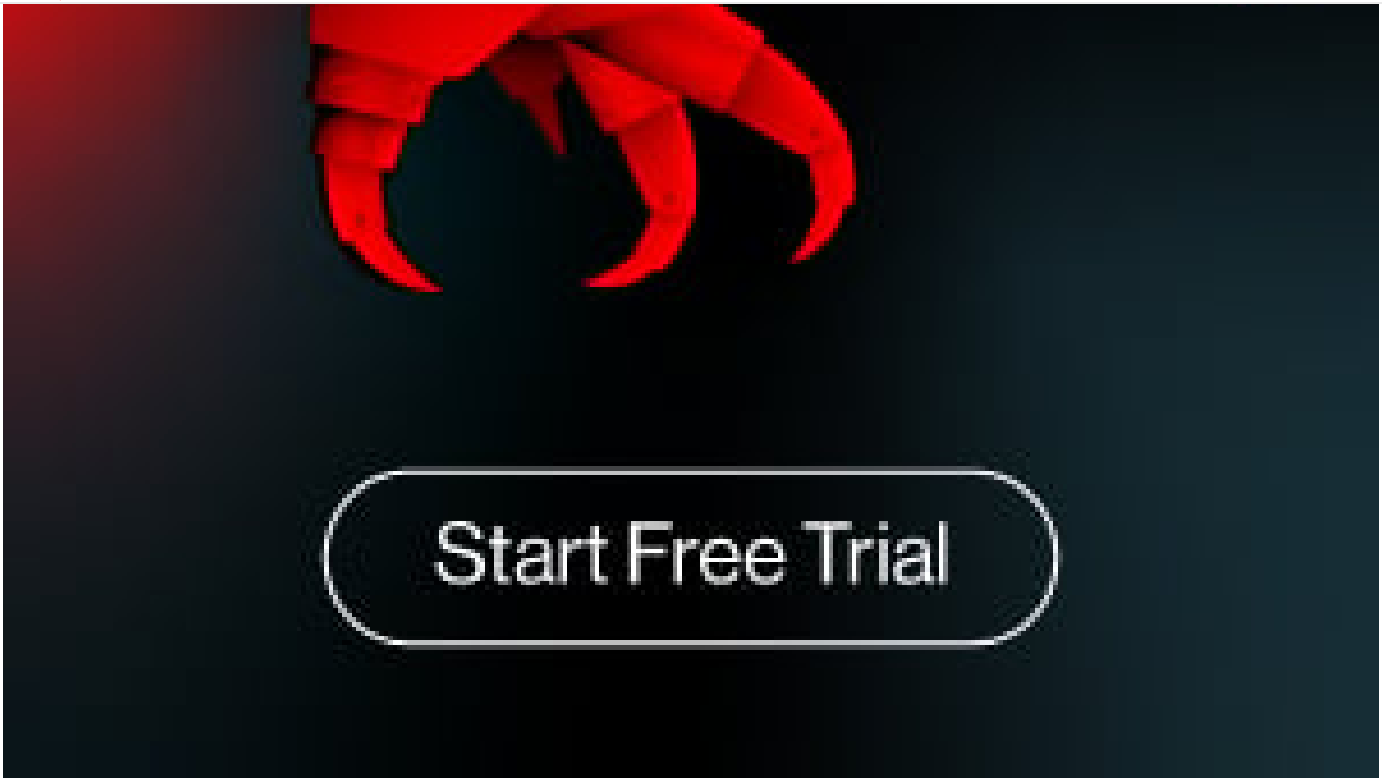
Featured
Recent
Video
Category
Start Free Trial

Incident Response Services

CONNECT WITH US

Featured
Recent
Video
Category
Start Free Trial





FEATURED ARTICLES

October 01, 2024

CrowdStrike Named a Leader in 2024 Gartner® Magic Quadrant™ for Endpoint Protection Platforms

September 25, 2024

Featured
Recent
Video
Category
Start Free Trial

SUBSCRIBE

« Falcon OverWatch Elite in Action: Tailored Threat Hunting Services Provide Individualized Care and Support

2022 Threat Hunting Report: Falcon OverWatch Looks Back to Prepare Defenders for Tomorrow's Adversaries »