Project Obsidian: Threat Hunting Kill Chain 3 - "The Logs are Gone?"

Abstract

"The Logs Are Gone? ...What do you mean The Logs Are Gone?"

What happens when an attacker clears the logs in an effort to hide their tracks?

Overview

What will we learn?

Here are the items we will cover:
- How to develop a Threat Hunting hypothesis
- Which sources we could use for developing a Threat Hunting hypothesis
- Research the methods attackers use to clear the Windows Event logs
- Find the data sources we need for our hunt
- Find the evidence of an attacker clearing the logs
- Answer the "How" of our hypothesis
- How to document your Threat Hunting using a simple template

A Note about Note-Taking during the Threat Hunting process

During, and throughout the Threat Hunting process, one should take good notes.

These notes will be **very** useful throughout the engagement.

Several times throughout this exercise I will be sure to mention that some fact or discovery we find should be added to the notes.

This is generally a good habit to embrace in the world of Information Security.

What is Threat Hunting?

```
"Threat hunting is ____(note: definition to be determined by the rgoup on Discord)"
```

Developing a Threat Hunting Hypothesis:

Question:

What guides the development of a Threat Hunting Hypothesis?

Here are some things to consider, and take note of when building a Threat Hunting hypothesis:

- The IT Environment
  - What are the Operating Systems in your environment? (Windows, Linux, Mac)
  - Does the org have Active Directory?
  - What data gets logged? (Windows Events, Sysmon, Zeek, etc.)
- Vulnerabilities
  - Does the org perform vulnerability scans?
  - If so, are the vulnerabilities analyzed & remediated?

    - Are there outstanding vulnerabilities that need to be remediated?

- MITRE ATT&CK Framework

    - Look at the MITRE ATT&CK Enterprise Matrix.
    - Take some time to look at the Enterprise tactics.
    - Within each of the tactics are techniques that attackers use.
    - These techniques can be used to help inspire your Threat Hunting adventures.

- Direct information from users/employees

    - What can your users/employees tell you about things they have seen in the environment?

FOR THE NOTES: *As the Threat Hunter, we would add the above items to our notes and continue.*

Threat Hunting Hypothesis:

In this scenario, we will consider the question:

What happens when an attacker clears the logs in an effort to hide their tracks?

We should start with a very broad hypothesis. Let's focus on the middle section of that question:

attacker clears the logs

Focusing on that, we could make a broad hypothesis like:

Broad Hypothesis:

```
Attackers will try to cover their tracks
```

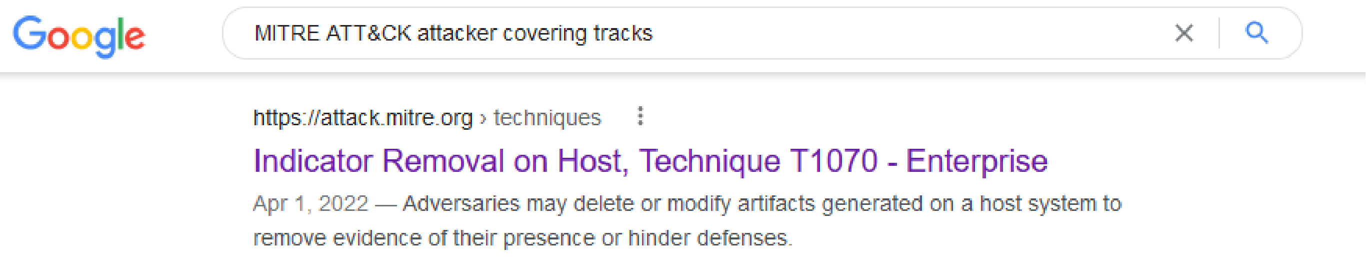
This hypothesis is broad because it only addresses the "Who" and the "What" - but not the "How"

FOR THE NOTES: *As the Threat Hunter, we would add our broad hypothesis to our notes and continue.*

So how can we develop this further into a specific hypothesis for our hunt?

Looking at our **NOTES**, we see that the MITRE ATT&CK Framework is listed as one of the things we could use to help build a Threat hunting hypothesis.

A quick Google search for `MITRE ATT&CK attacker covering tracks` brings us to the following MITRE ATT&CK page:

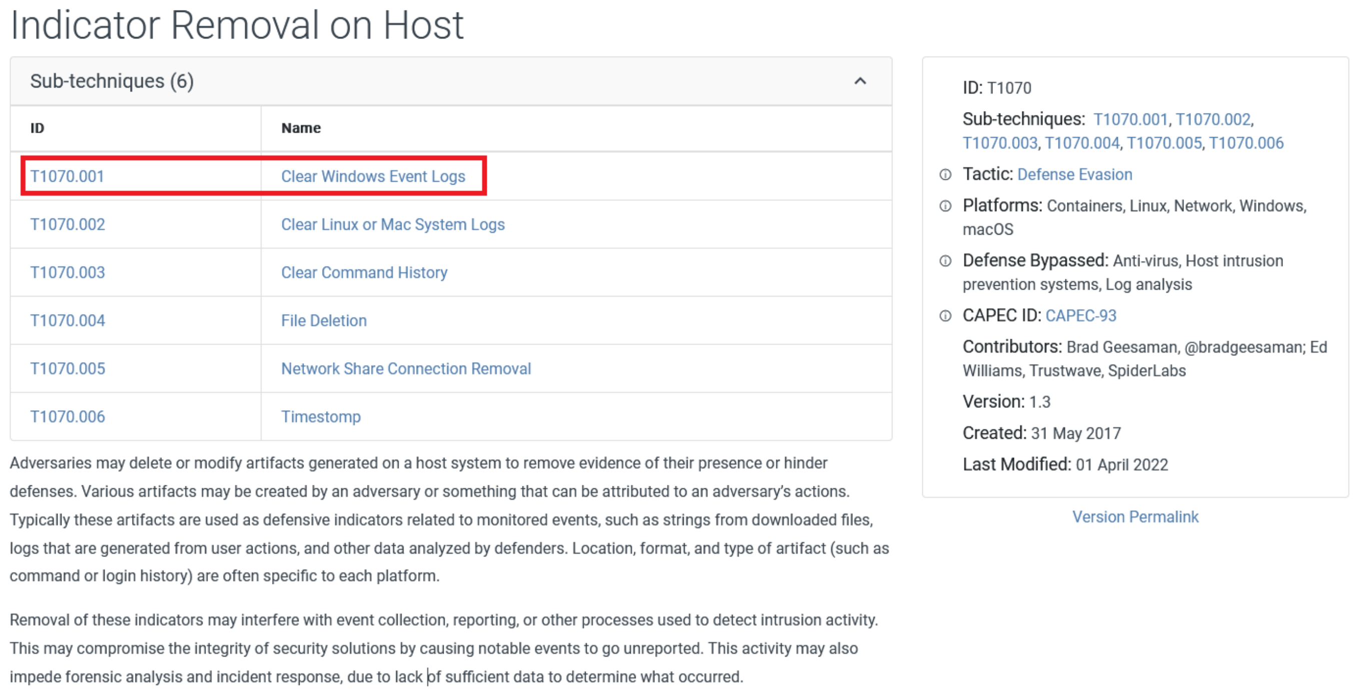


https://attack.mitre.org › techniques

Indicator Removal on Host, Technique T1070 - Enterprise

Apr 1, 2022 — Adversaries may delete or modify artifacts generated on a host system to remove evidence of their presence or hinder defenses.

Let's follow the link to read what this MITRE ATT&CK article says.



Indicator Removal on Host

Sub-techniques (6)

ID	Name
T1070.001	Clear Windows Event Logs
T1070.002	Clear Linux or Mac System Logs
T1070.003	Clear Command History
T1070.004	File Deletion
T1070.005	Network Share Connection Removal
T1070.006	Timestomp

Adversaries may delete or modify artifacts generated on a host system to remove evidence of their presence or hinder defenses. Various artifacts may be created by an adversary or something that can be attributed to an adversary's actions. Typically these artifacts are used as defensive indicators related to monitored events, such as strings from downloaded files, logs that are generated from user actions, and other data analyzed by defenders. Location, format, and type of artifact (such as command or login history) are often specific to each platform.

Removal of these indicators may interfere with event collection, reporting, or other processes used to detect intrusion activity. This may compromise the integrity of security solutions by causing notable events to go unreported. This activity may also impede forensic analysis and incident response, due to lack of sufficient data to determine what occurred.

ID: T1070

Sub-techniques: T1070.001, T1070.002, T1070.003, T1070.004, T1070.005, T1070.006

Tactic: Defense Evasion

Platforms: Containers, Linux, Network, Windows, macOS

Defense Bypassed: Anti-virus, Host intrusion prevention systems, Log analysis

CAPEC ID: CAPEC-93

Contributors: Brad Geesaman, @bradgeesaman; Ed Williams, Trustwave, SpiderLabs

Version: 1.3

Created: 31 May 2017

Last Modified: 01 April 2022

Version Permalink

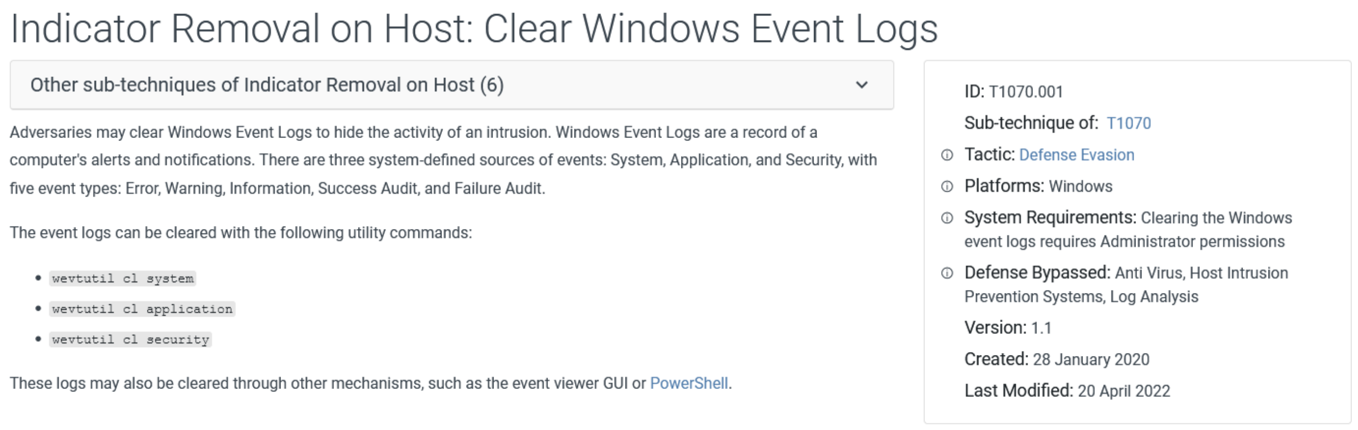
Source: https://attack.mitre.org/techniques/T1070/

In this scenario, we are looking for information about Windows Event logs being cleared.

We can clearly see something more relevant to our scenario in the Sub-Technique listed above:

T1070.001 "Clear Windows Event Logs"

Let's take a look at the details of the MITRE ATT&CK Sub-Technique T1070.001



Indicator Removal on Host: Clear Windows Event Logs

Other sub-techniques of Indicator Removal on Host (6)

Adversaries may clear Windows Event Logs to hide the activity of an intrusion. Windows Event Logs are a record of a computer's alerts and notifications. There are three system-defined sources of events: System, Application, and Security, with five event types: Error, Warning, Information, Success Audit, and Failure Audit.

The event logs can be cleared with the following utility commands:

- `wevtutil cl system`
- `wevtutil cl application`
- `wevtutil cl security`

These logs may also be cleared through other mechanisms, such as the event viewer GUI or PowerShell.

ID: T1070.001

Sub-technique of: T1070

Tactic: Defense Evasion

Platforms: Windows

System Requirements: Clearing the Windows event logs requires Administrator permissions

Defense Bypassed: Anti Virus, Host Intrusion Prevention Systems, Log Analysis

Version: 1.1

Created: 28 January 2020

Last Modified: 20 April 2022

Source: https://attack.mitre.org/techniques/T1070/001/

Note the methods listed above in T1070.001

- Windows Command `wevtutil cl system`
- Windows Command `wevtutil cl application`
- Windows Command `wevtutil cl security`
- Event Viewer GUI
- PowerShell

FOR THE NOTES: *As the Threat Hunter, we would add the above MITRE ATT&CK technique T1070, the sub-technique T1070.001, the attack methods listed, and any Source URLs to our notes and continue.*

Using the information from the MITRE ATT&CK Sub-Technique T1070.001 about the specific methods attackers will use to cover their tracks, we can update our Threat Hunting Hypothesis to be more specific.

A QUICK REVIEW OF HOW WE GOT HERE:

We started with a broad hypothesis, and then we did some research & found some details from the MITRE ATT&CK framework to enhance our broad hypothesis into a specific hypothesis:

Before:

Broad Hypothesis:

```
Attackers will try to cover their tracks
```

After:

Specific Hypothesis:

``` Attackers will try to cover their tracks using one of the following methods: - The Event Viewer GUI - Windows Commands (such as "wevtutil cl") - PowerShell ```

FOR THE NOTES: *As the Threat Hunter, we would add our specific hypothesis to our notes and continue.*

Where can we find the data needed for our hypothesis?

It's important to know your DATA SOURCES when conducting a Threat Hunt!

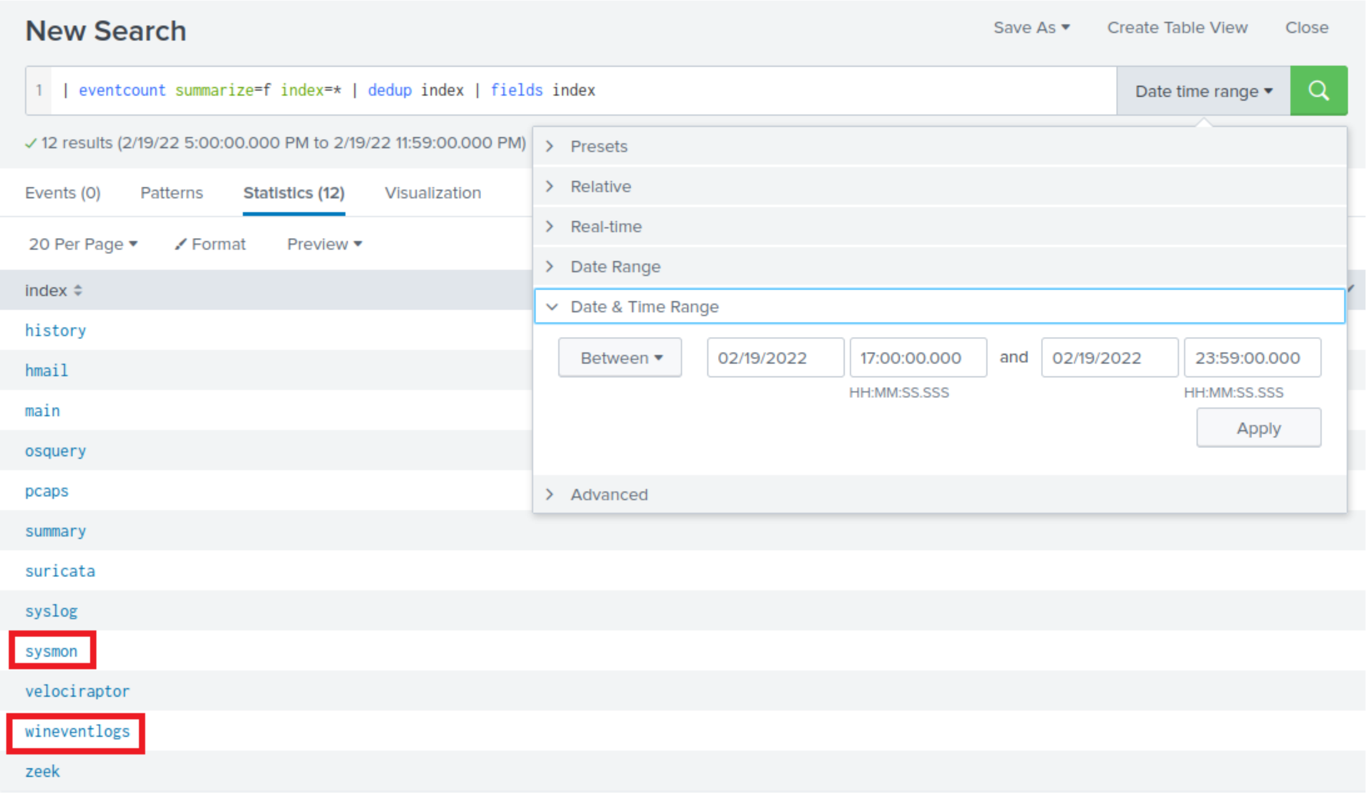
Let's take a look at the data sources available to us in Splunk.

What this Splunk query we are about to run basically does is to search the Splunk events (`eventcount`), split the events by index (`summarize=false`), search for any index (`index=*`), de-dupe the results by the field "index" (`dedupe index`), and then use `fields index` to only show the "index" field.

Set date/time between 02/19/22 17:00 to 02/19/22 23:59:

Splunk Search:

```
| eventcount summarize=f index=* | dedup index | fields index
```



```
1 | eventcount summarize=f index=* | dedup index | fields index
```

✓ 12 results (2/19/22 5:00:00.000 PM to 2/19/22 11:59:00.000 PM)

Date & Time Range: Between 02/19/2022 17:00:00.000 and 02/19/2022 23:59:00.000

index
history
hmail
main
osquery
pcaps
summary
suricata
syslog
sysmon
velociraptor
wineventlogs
zeek

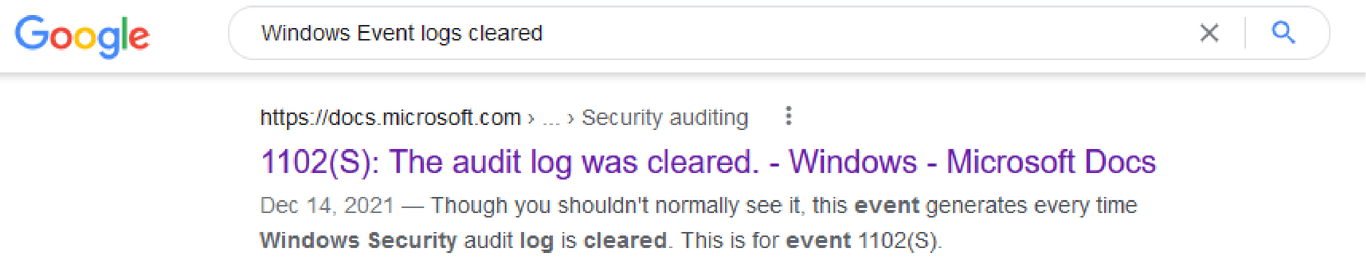
Source for above Splunk Query: https://gist.github.com/jonathanhle/fceaae49fc207649b3be930e3c46f2ad

Within the list of data sources (indexes) above, we will find what we need within the **Sysmon** and **Wineventlogs** indexes because these data sources will show us what was happening INSIDE of a host.

FOR THE NOTES: *As the Threat Hunter, we would add the Splunk indexes **Sysmon** and **Wineventlogs** as "Data Sources" to our notes and continue.*

To begin our hunt, we should look for evidence that the Windows Event logs were actually cleared.

A simple Google search for `Windows Event logs cleared` gives us a link to a Microsoft article about Windows Security Event ID 1102.



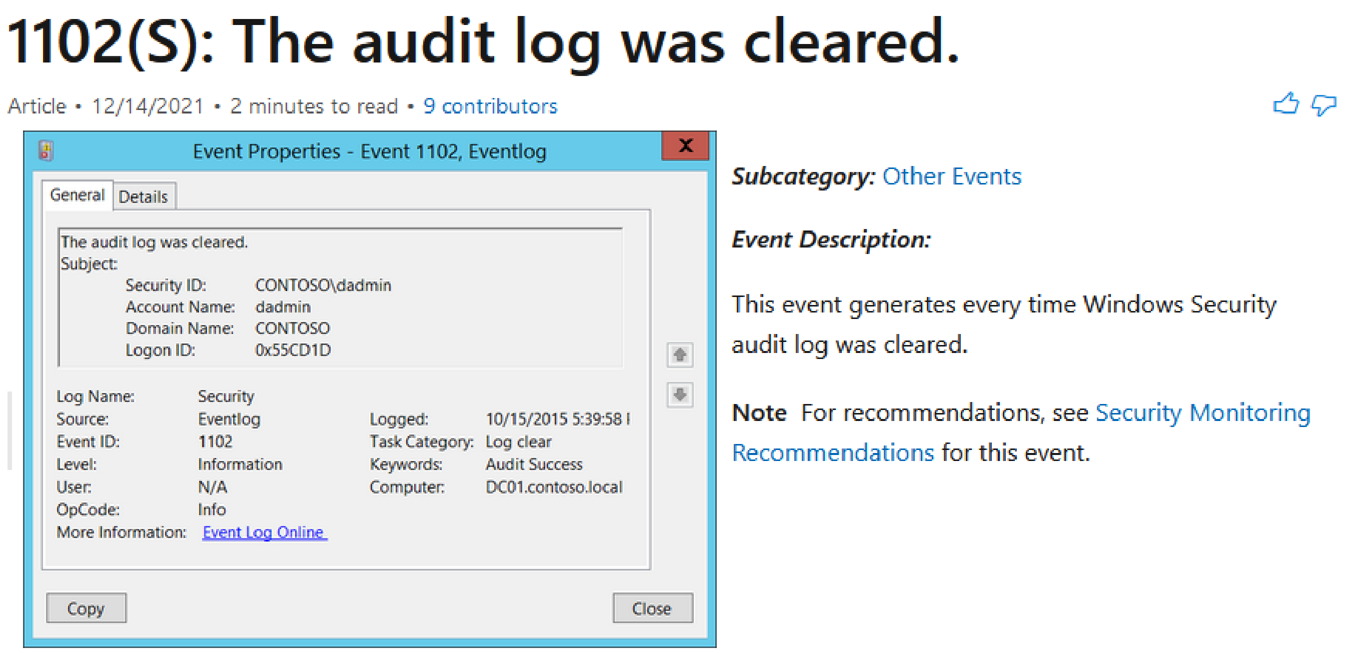
Windows Event logs cleared

https://docs.microsoft.com › ... › Security auditing

1102(S): The audit log was cleared. - Windows - Microsoft Docs

Dec 14, 2021 — Though you shouldn't normally see it, this **event** generates every time **Windows Security** audit **log** is **cleared**. This is for **event** 1102(S).

Let's take a look at this Microsoft article about Windows Security Event ID 1102.



1102(S): The audit log was cleared.

Article • 12/14/2021 • 2 minutes to read • 9 contributors

Event Properties - Event 1102, Eventlog

```
The audit log was cleared.
Subject:
        Security ID:      CONTOSO\dadmin
        Account Name:     dadmin
        Domain Name:      CONTOSO
        Logon ID:         0x55CD1D

Log Name:          Security
Source:            Eventlog           Logged:          10/15/2015 5:39:58 I
Event ID:          1102               Task Category:   Log clear
Level:             Information        Keywords:        Audit Success
User:              N/A                Computer:        DC01.contoso.local
OpCode:            Info
More Information:  Event Log Online
```

Subcategory: Other Events

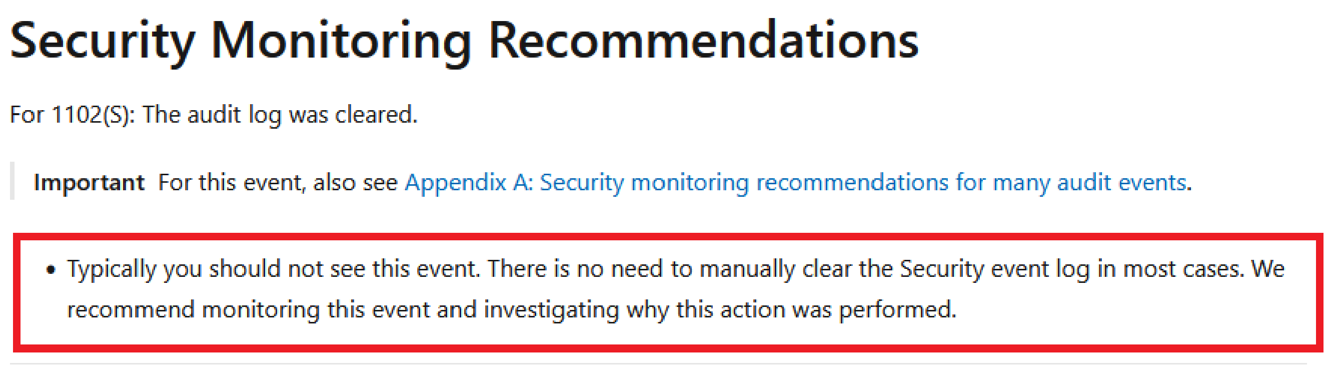
Event Description:

This event generates every time Windows Security audit log was cleared.

Note For recommendations, see Security Monitoring Recommendations for this event.

Source: https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-1102

In this article, pay close attention to what the Microsoft article says for "Security Monitoring Recommendations."



Security Monitoring Recommendations

For 1102(S): The audit log was cleared.

> **Important** For this event, also see Appendix A: Security monitoring recommendations for many audit events.

- Typically you should not see this event. There is no need to manually clear the Security event log in most cases. We recommend monitoring this event and investigating why this action was performed.

FOR THE NOTES: *As the Threat Hunter, we would add the information about Windows Security Event ID 1102 (including any Source URLs) to our notes and continue.*

Let's go search for this event within our Splunk data.

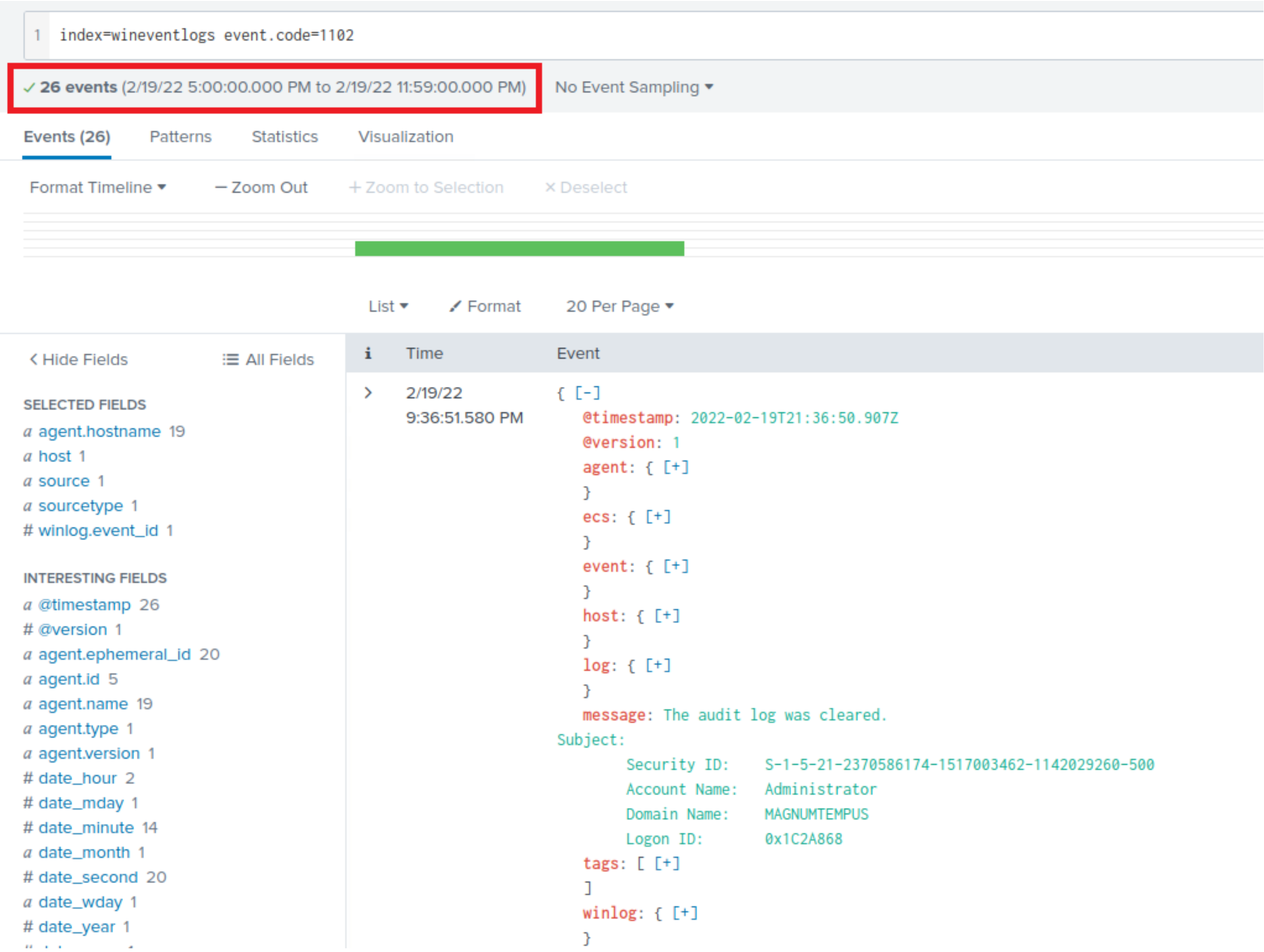
According to our **NOTES**, we have **Wineventlogs** (which contains the typical Windows Event logs found on a Windows Operating System) listed as a data source in Splunk.

Let's search the `wineventlogs` index in Splunk for `event.code` 1102.

Set date/time between 02/19/22 17:00 to 02/19/22 23:59:

Splunk Search:

```
index=wineventlogs event.code=1102
```



```
index=wineventlogs event.code=1102
```

✓ 26 events (2/19/22 5:00:00.000 PM to 2/19/22 11:59:00.000 PM)

SELECTED FIELDS
- a agent.hostname 19
- a host 1
- a source 1
- a sourcetype 1
- # winlog.event_id 1

INTERESTING FIELDS
- a @timestamp 26
- # @version 1
- a agent.ephemeral_id 20
- a agent.id 5
- a agent.name 19
- a agent.type 1
- a agent.version 1
- # date_hour 2
- # date_mday 1
- # date_minute 14
- a date_month 1
- # date_second 20
- a date_wday 1
- # date_year 1

2/19/22 9:36:51.580 PM

```
{ [-]
   @timestamp: 2022-02-19T21:36:50.907Z
   @version: 1
   agent: { [+]
   }
   ecs: { [+]
   }
   event: { [+]
   }
   host: { [+]
   }
   log: { [+]
   }
   message: The audit log was cleared.
Subject:
          Security ID:    S-1-5-21-2370586174-1517003462-1142029260-500
          Account Name:   Administrator
          Domain Name:    MAGNUMTEMPUS
          Logon ID:       0x1C2A868
   tags: [ [+]
   ]
   winlog: { [+]
   }
```

Indeed! 26 events discovered showing proof that the Windows Event logs were cleared!

FOR THE NOTES: *As the Threat Hunter, we would add this broad Splunk query (including the date/time specified for the query), as well as some details about the results to our notes and continue.*

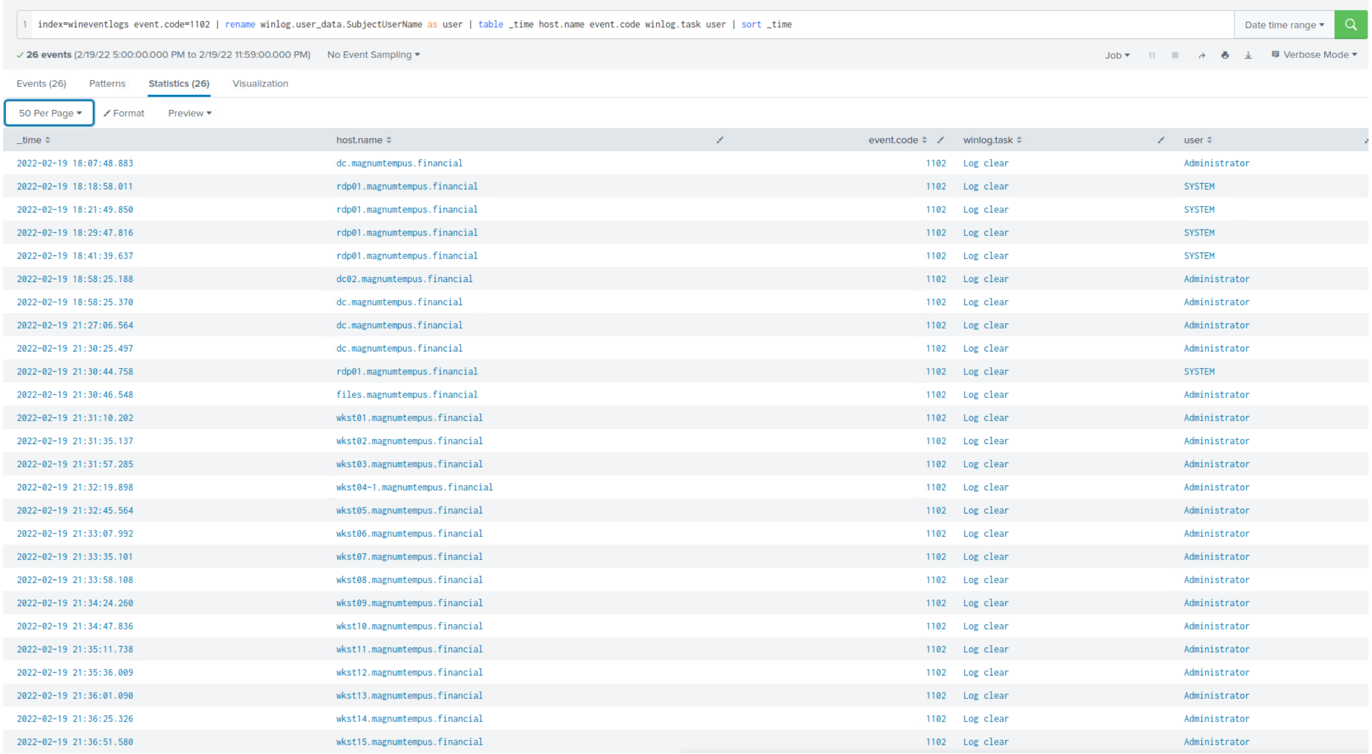
Let's alter that Splunk query to make it look more presentable.

We will search the `wineventlogs` index for `event.code` 1102, and we will use `rename` to rename the long field `winlog.user_data.SubjectUserName` as `user` (which is much nicer, right?), and then we will arrange the results of the query using `table` and we will specify the fields we want displayed (`_time host.name event.code winlog.task user`), and then we will `sort` the results by `_time` (from oldest at the top to newest at the bottom).

Set date/time between 02/19/22 17:00 to 02/19/22 23:59:

Splunk Search:

```
index=wineventlogs event.code=1102 | rename winlog.user_data.SubjectUserName as user | table _time host.name event.code winlog.task user | sort _time
```



```
index=wineventlogs event.code=1102 | rename winlog.user_data.SubjectUserName as user | table _time host.name event.code winlog.task user | sort _time
```

✓ **26 events** (2/19/22 5:00:00.000 PM to 2/19/22 11:59:00.000 PM)

_time	host.name	event.code	winlog.task	user
2022-02-19 18:07:48.883	dc.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 18:18:58.011	rdp01.magnumtempus.financial	1102	Log clear	SYSTEM
2022-02-19 18:21:49.850	rdp01.magnumtempus.financial	1102	Log clear	SYSTEM
2022-02-19 18:29:47.816	rdp01.magnumtempus.financial	1102	Log clear	SYSTEM
2022-02-19 18:41:39.637	rdp01.magnumtempus.financial	1102	Log clear	SYSTEM
2022-02-19 18:58:25.188	dc02.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 18:58:25.370	dc.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:27:06.564	dc.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:30:25.497	dc.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:30:44.758	rdp01.magnumtempus.financial	1102	Log clear	SYSTEM
2022-02-19 21:30:46.548	files.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:31:10.202	wkst01.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:31:35.137	wkst02.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:31:57.285	wkst03.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:32:19.898	wkst04-1.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:32:45.564	wkst05.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:33:07.992	wkst06.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:33:35.101	wkst07.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:33:58.108	wkst08.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:34:24.260	wkst09.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:34:47.836	wkst10.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:35:11.738	wkst11.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:35:36.009	wkst12.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:36:01.090	wkst13.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:36:25.326	wkst14.magnumtempus.financial	1102	Log clear	Administrator
2022-02-19 21:36:51.580	wkst15.magnumtempus.financial	1102	Log clear	Administrator

Wow! Interesting results!!

Here we can see evidence of the Windows Event logs being cleared on several hosts!

Also, it's interesting to note that the search results did not specifically show if the Windows Event Logs were cleared via the Event Viewer GUI, Windows Command ("wevtutil cl"), or PowerShell.

In any case, we see that the Windows Event logs *were* cleared.

FOR THE NOTES: *As the Threat Hunter, we would add this specific Splunk query (including the date/time specified for the query), as well as some details about the results to our notes and continue.*

So, are we done?

Did we accomplish what we set out to do?

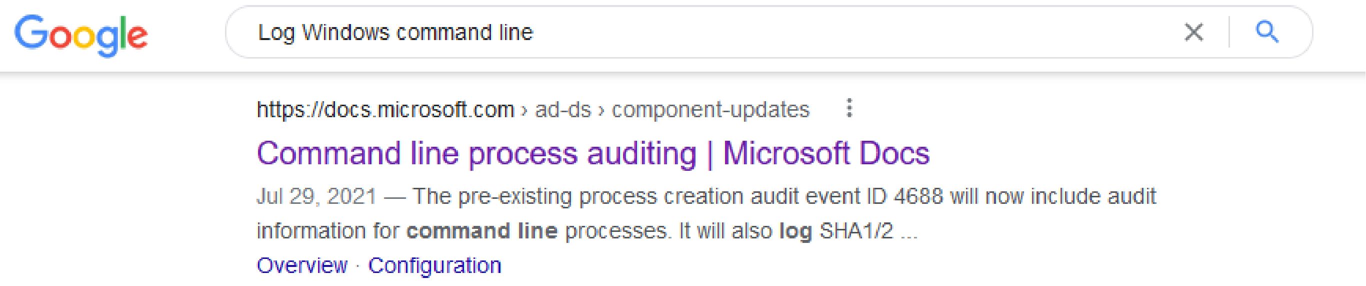
Some would say "yes" (and they're not entirely wrong--we did find a way to detect when the Windows Event Logs get cleared), but what if we wanted to find the source of this activity? What if we wanted to find which command (or commands) the attacker may have used to clear these logs? What if we wanted to find the answer to "How" the attackers cleared the Windows Event logs?

Let's explore further to answer those questions.

Going back to our **NOTES**, we can see in our specific hypothesis that one of the methods attackers use to cover their tracks is by using the "wevtutil cl" command. We should search for this command in our Splunk data.

But how would we do that? How could we find specific Windows commands that were executed on a host?

Let's just Google `Log Windows command line` and see what it gives us.



Log Windows command line

https://docs.microsoft.com › ad-ds › component-updates

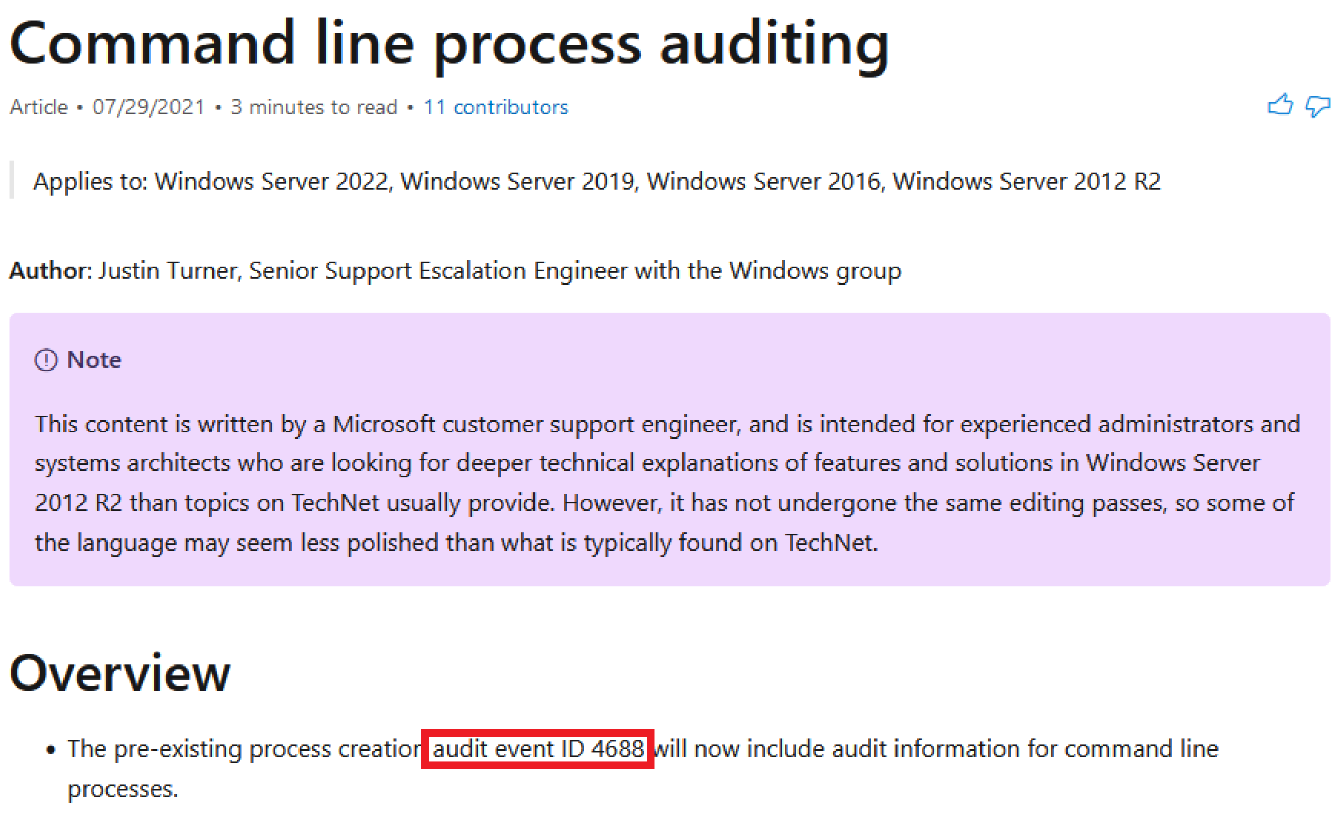
Command line process auditing | Microsoft Docs

Jul 29, 2021 — The pre-existing process creation audit event ID 4688 will now include audit information for **command line** processes. It will also **log** SHA1/2 ...

Overview · Configuration

One of the first results is a Microsoft article: "Command line process auditing"

Let's take a look at that link.



Command line process auditing

Article • 07/29/2021 • 3 minutes to read • 11 contributors

> Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2

Author: Justin Turner, Senior Support Escalation Engineer with the Windows group

Note

This content is written by a Microsoft customer support engineer, and is intended for experienced administrators and systems architects who are looking for deeper technical explanations of features and solutions in Windows Server 2012 R2 than topics on TechNet usually provide. However, it has not undergone the same editing passes, so some of the language may seem less polished than what is typically found on TechNet.

Overview

- The pre-existing process creation audit event ID 4688 will now include audit information for command line processes.

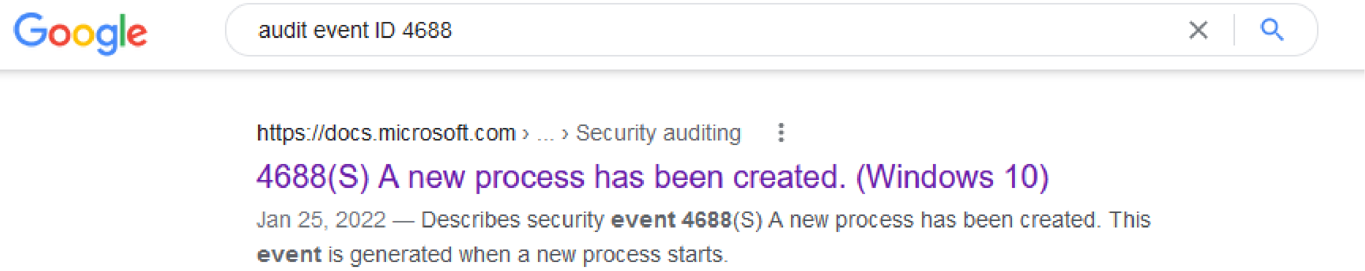
Source: https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/manage/component-updates/command-line-process-auditing

Reading the Microsoft article, it mentions **"audit event ID 4688"**

What is this "audit event ID 4688?"

Let's find out.

A quick Google search for `audit event ID 4688` leads us to an article titled "4688(S) A new process has been created"

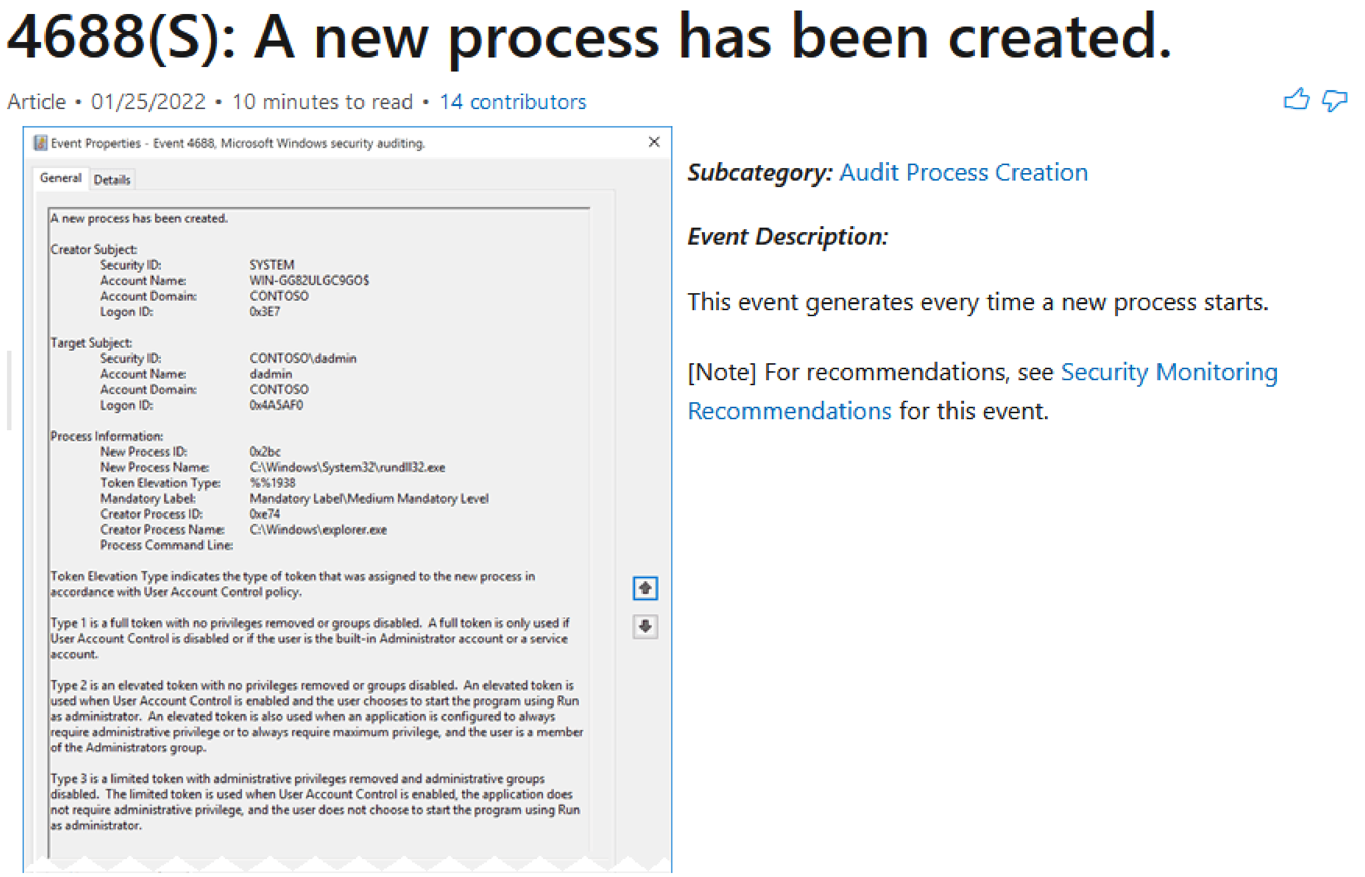


https://docs.microsoft.com › ... › Security auditing

4688(S) A new process has been created. (Windows 10)

Jan 25, 2022 — Describes security **event 4688**(S) A new process has been created. This **event** is generated when a new process starts.

Let's take a look at that article.



4688(S): A new process has been created.

Article • 01/25/2022 • 10 minutes to read • 14 contributors

Event Properties - Event 4688, Microsoft Windows security auditing.

```
A new process has been created.

Creator Subject:
        Security ID:            SYSTEM
        Account Name:           WIN-GG82ULGC9GO$
        Account Domain:         CONTOSO
        Logon ID:               0x3E7

Target Subject:
        Security ID:            CONTOSO\dadmin
        Account Name:           dadmin
        Account Domain:         CONTOSO
        Logon ID:               0x4A5AF0

Process Information:
        New Process ID:         0x2bc
        New Process Name:       C:\Windows\System32\rundll32.exe
        Token Elevation Type:   %%1938
        Mandatory Label:        Mandatory Label\Medium Mandatory Level
        Creator Process ID:     0xe74
        Creator Process Name:   C:\Windows\explorer.exe
        Process Command Line:

Token Elevation Type indicates the type of token that was assigned to the new process in
accordance with User Account Control policy.

Type 1 is a full token with no privileges removed or groups disabled.  A full token is only used if
User Account Control is disabled or if the user is the built-in Administrator account or a service
account.

Type 2 is an elevated token with no privileges removed or groups disabled.  An elevated token is
used when User Account Control is enabled and the user chooses to start the program using Run
as administrator.  An elevated token is also used when an application is configured to always
require administrative privilege or to always require maximum privilege, and the user is a member
of the Administrators group.

Type 3 is a limited token with administrative privileges removed and administrative groups
disabled.  The limited token is used when User Account Control is enabled, the application does
not require administrative privilege, and the user does not choose to start the program using Run
as administrator.
```

Subcategory: Audit Process Creation

Event Description:

This event generates every time a new process starts.

[Note] For recommendations, see Security Monitoring Recommendations for this event.

Source: https://docs.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4688

One of the fields we see in Windows Security Event ID 4688 is the "Process Command Line" field.

Great!

So it appears that we can search Splunk for the Windows Security Event ID 4688, and specify the value we are looking for in the "Process Command Line" field!

Right?

FOR THE NOTES: *As the Threat Hunter, we would add this information we have found about Windows Security Event ID 4688 (including any Source URLs) to our notes and continue.*

Broad Search Queries:

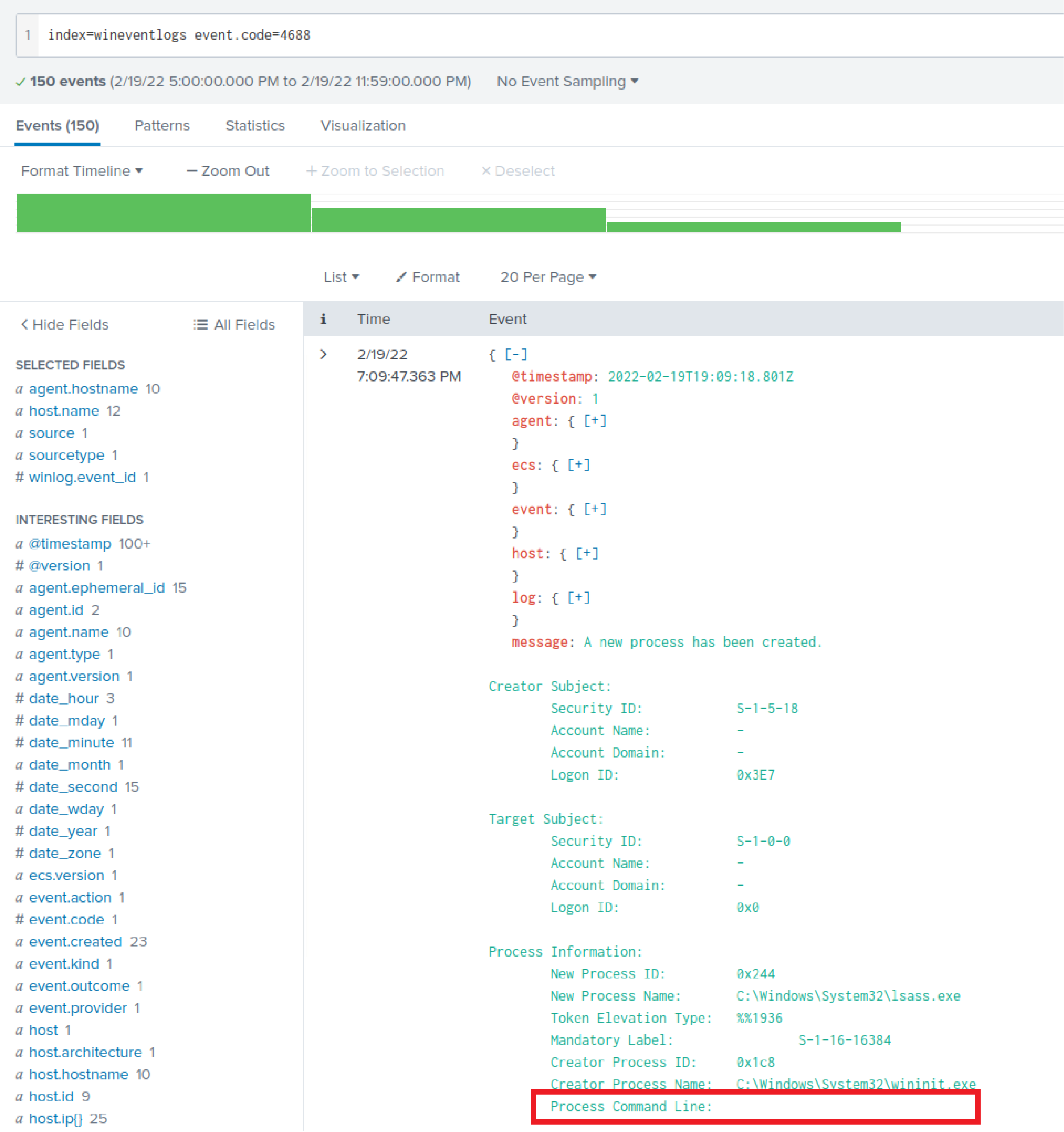
Let's take a look at Windows Security Event ID 4688 in Splunk and see what the "Process Command Line" field will show us.

We will search the `wineventlogs` index for `event.code` 4688.

Set date/time between 02/19/22 17:00 to 02/19/22 23:59:

Splunk Search:

```
index=wineventlogs event.code=4688
```



```
index=wineventlogs event.code=4688
```

✓ **150 events** (2/19/22 5:00:00.000 PM to 2/19/22 11:59:00.000 PM) No Event Sampling ▾

SELECTED FIELDS

a	agent.hostname	10
a	host.name	12
a	source	1
a	sourcetype	1
#	winlog.event_id	1

INTERESTING FIELDS

a	@timestamp	100+
#	@version	1
a	agent.ephemeral_id	15
a	agent.id	2
a	agent.name	10
a	agent.type	1
a	agent.version	1
#	date_hour	3
#	date_mday	1
#	date_minute	11
a	date_month	1
#	date_second	15
a	date_wday	1
#	date_year	1
#	date_zone	1
a	ecs.version	1
a	event.action	1
#	event.code	1
a	event.created	23
a	event.kind	1
a	event.outcome	1
a	event.provider	1
a	host	1
a	host.architecture	1
a	host.hostname	10
a	host.id	9
a	host.ip{}	25

2/19/22 7:09:47.363 PM

```
{ [-]
   @timestamp: 2022-02-19T19:09:18.801Z
   @version: 1
   agent: { [+]
   }
   ecs: { [+]
   }
   event: { [+]
   }
   host: { [+]
   }
   log: { [+]
   }
   message: A new process has been created.

Creator Subject:
        Security ID:            S-1-5-18
        Account Name:           -
        Account Domain:         -
        Logon ID:               0x3E7

Target Subject:
        Security ID:            S-1-0-0
        Account Name:           -
        Account Domain:         -
        Logon ID:               0x0

Process Information:
        New Process ID:         0x244
        New Process Name:       C:\Windows\System32\lsass.exe
        Token Elevation Type:   %%1936
        Mandatory Label:                S-1-16-16384
        Creator Process ID:     0x1c8
        Creator Process Name:   C:\Windows\System32\wininit.exe
        Process Command Line:
```

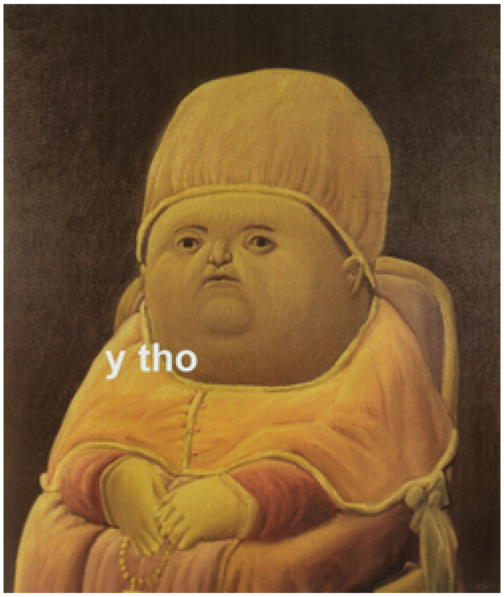
Bummer!

Looking at the results, we don't find any data for the "Process Command Line" field in Windows Security Event ID 4688.

FOR THE NOTES: *As the Threat Hunter, we would add this specific Splunk query (including the date/time specified for the query), as well as some details about the results to our notes and continue.*

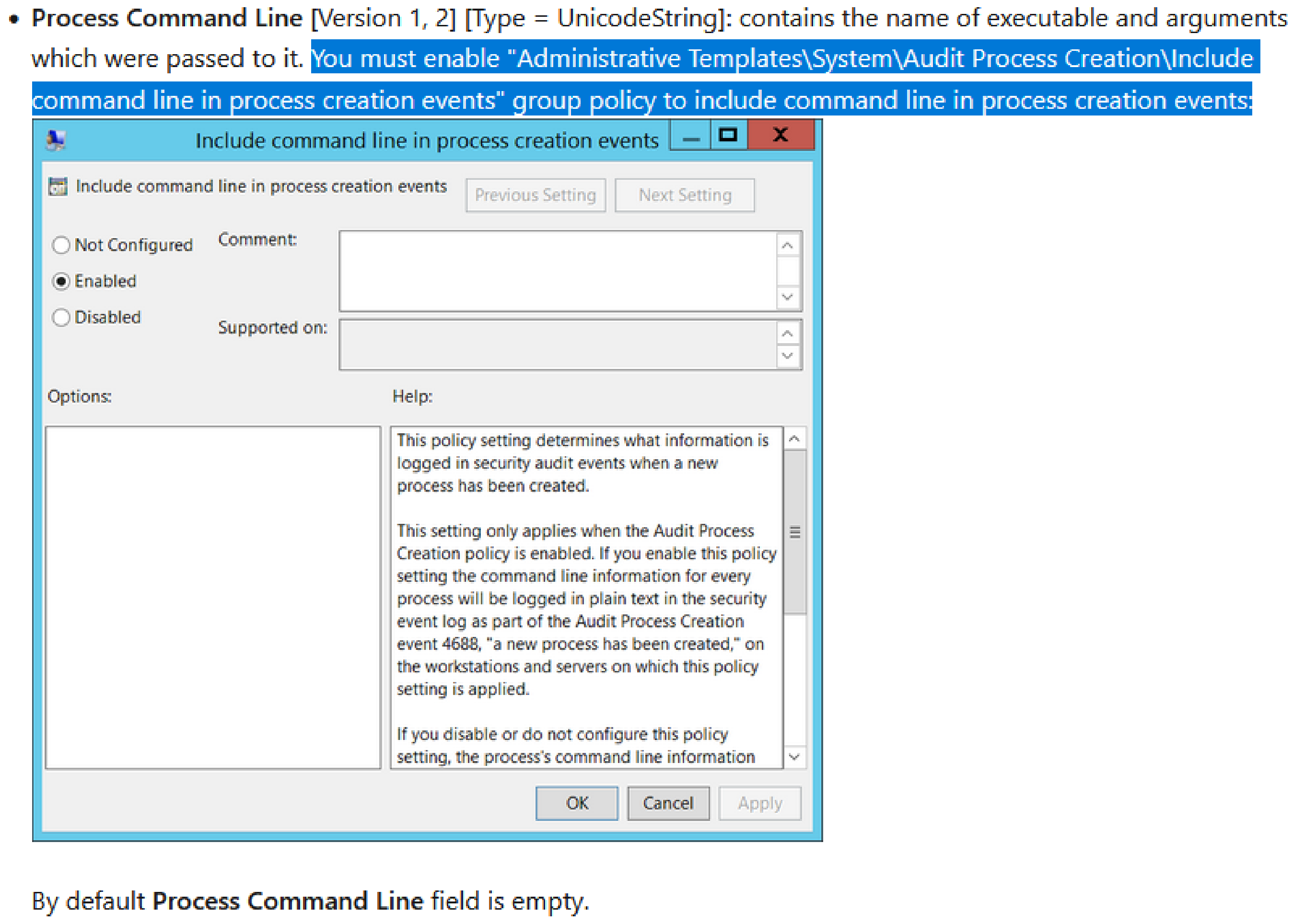
So we didn't find any Command Line data in Windows Security Event ID 4688.

Why?



According to the Microsoft article previously mentioned ("4688(S): A new process has been created"):

"You must enable 'Administrative Templates|System|Audit Process Creation|Include command line in process creation events' group policy to include command line in process creation events"



- **Process Command Line** [Version 1, 2] [Type = UnicodeString]: contains the name of executable and arguments which were passed to it. You must enable "Administrative Templates\System\Audit Process Creation\Include command line in process creation events" group policy to include command line in process creation events:

Include command line in process creation events
Include command line in process creation events
Previous Setting
Next Setting

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on:

Options:

Help:

This policy setting determines what information is logged in security audit events when a new process has been created.

This setting only applies when the Audit Process Creation policy is enabled. If you enable this policy setting the command line information for every process will be logged in plain text in the security event log as part of the Audit Process Creation event 4688, "a new process has been created," on the workstations and servers on which this policy setting is applied.

If you disable or do not configure this policy setting, the process's command line information

By default **Process Command Line** field is empty.

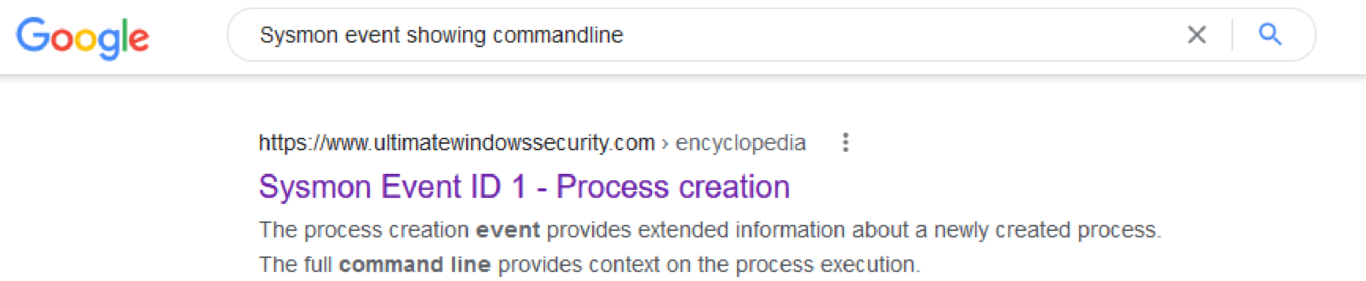
So are we dead in the water?
Is there nothing else we can do to find what commands the attacker used to clear the logs?

Checking our **NOTES** that we took, recall that we made note of the two data sources (**Sysmon** and **Wineventlogs**) in Splunk earlier.

Let's take a look at the **Sysmon** data.

Question:
Which Sysmon Event ID could possibly show us the command line details?

A quick Google search for `Sysmon event showing commandline` leads to an article about "Sysmon Event ID 1 - Process Creation."



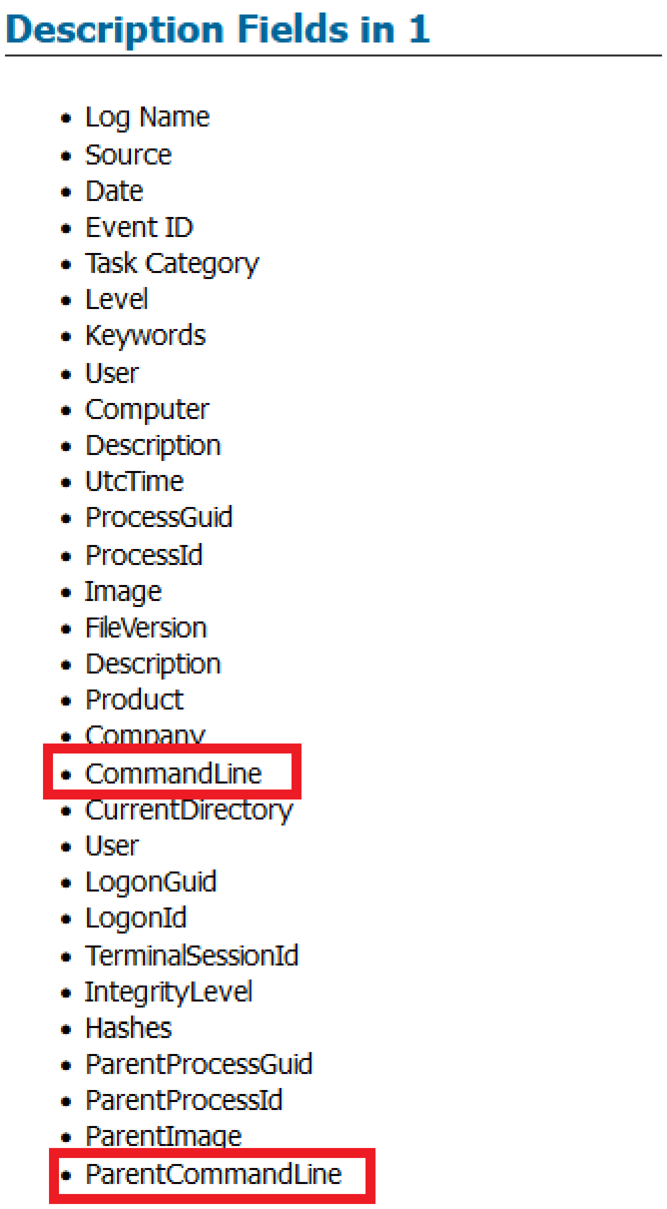
Sysmon event showing commandline

https://www.ultimatewindowssecurity.com › encyclopedia

Sysmon Event ID 1 - Process creation

The process creation **event** provides extended information about a newly created process. The full **command line** provides context on the process execution.

We see in that article that it lists the fields in Sysmon Event ID 1:



Description Fields in 1

- Log Name
- Source
- Date
- Event ID
- Task Category
- Level
- Keywords
- User
- Computer
- Description
- UtcTime
- ProcessGuid
- ProcessId
- Image
- FileVersion
- Description
- Product
- Company
- CommandLine
- CurrentDirectory
- User
- LogonGuid
- LogonId
- TerminalSessionId
- IntegrityLevel
- Hashes
- ParentProcessGuid
- ParentProcessId
- ParentImage
- ParentCommandLine

Source: https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/event.aspx?eventid=90001

FOR THE NOTES: *As the Threat Hunter, we would add this information we have discovered about Sysmon Event ID 1 (including any Source URLs) to our notes and continue.*

All is not lost!!

We have Sysmon Event ID 1 ("Process Creation") which shows us the Command Line, as well as the Parent Command Line!

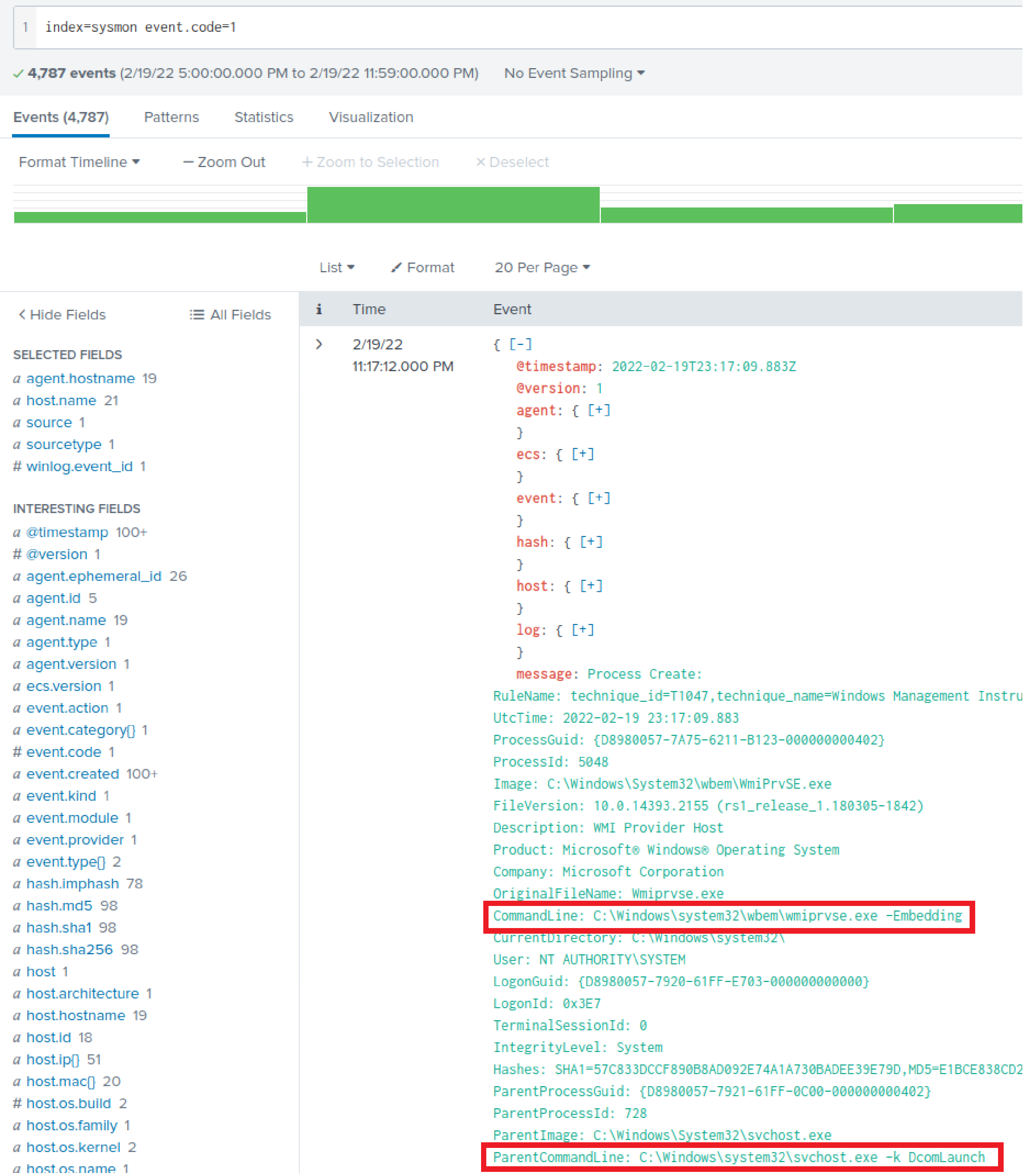
Let's take a look at Sysmon Event ID 1 in Splunk. We will search the `sysmon` index for `event.code` 1.

IMPORTANT NOTE: This is just a simple Splunk query to see what the command line data looks like from Sysmon Event ID 1. The actual values in the command line data in these results don't really matter for this search. We just want to VERIFY that we can see command line data.

Set date/time between 02/19/22 17:00 to 02/19/22 23:59:

Splunk Search:

```
index=sysmon event.code=1
```



```
1  index=sysmon event.code=1
```

✓ **4,787 events** (2/19/22 5:00:00.000 PM to 2/19/22 11:59:00.000 PM)   No Event Sampling ▾

SELECTED FIELDS
- a agent.hostname 19
- a host.name 21
- a source 1
- a sourcetype 1
- # winlog.event_id 1

INTERESTING FIELDS
- a @timestamp 100+
- # @version 1
- a agent.ephemeral_id 26
- a agent.id 5
- a agent.name 19
- a agent.type 1
- a agent.version 1
- a ecs.version 1
- a event.action 1
- a event.category{} 1
- # event.code 1
- a event.created 100+
- a event.kind 1
- a event.module 1
- a event.provider 1
- a event.type{} 2
- a hash.imphash 78
- a hash.md5 98
- a hash.sha1 98
- a hash.sha256 98
- a host 1
- a host.architecture 1
- a host.hostname 19
- a host.id 18
- a host.ip{} 51
- a host.mac{} 20
- # host.os.build 2
- a host.os.family 1
- a host.os.kernel 2
- a host.os.name 1

2/19/22
11:17:12.000 PM

```
{ [-]
   @timestamp: 2022-02-19T23:17:09.883Z
   @version: 1
   agent: { [+]
   }
   ecs: { [+]
   }
   event: { [+]
   }
   hash: { [+]
   }
   host: { [+]
   }
   log: { [+]
   }
   message: Process Create:
RuleName: technique_id=T1047,technique_name=Windows Management Instru
UtcTime: 2022-02-19 23:17:09.883
ProcessGuid: {D8980057-7A75-6211-B123-000000000402}
ProcessId: 5048
Image: C:\Windows\System32\wbem\WmiPrvSE.exe
FileVersion: 10.0.14393.2155 (rs1_release_1.180305-1842)
Description: WMI Provider Host
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: Wmiprvse.exe
CommandLine: C:\Windows\system32\wbem\wmiprvse.exe -Embedding
CurrentDirectory: C:\Windows\system32\
User: NT AUTHORITY\SYSTEM
LogonGuid: {D8980057-7920-61FF-E703-000000000000}
LogonId: 0x3E7
TerminalSessionId: 0
IntegrityLevel: System
Hashes: SHA1=57C833DCCF890B8AD092E74A1A730BADEE39E79D,MD5=E1BCE838CD2
ParentProcessGuid: {D8980057-7921-61FF-0C00-000000000402}
ParentProcessId: 728
ParentImage: C:\Windows\System32\svchost.exe
ParentCommandLine: C:\Windows\system32\svchost.exe -k DcomLaunch
```

Great! Now that we have verified that we can see the command line values in Sysmon Event ID 1, let's search for the command "wevtutil cl" listed in our specific hypothesis.

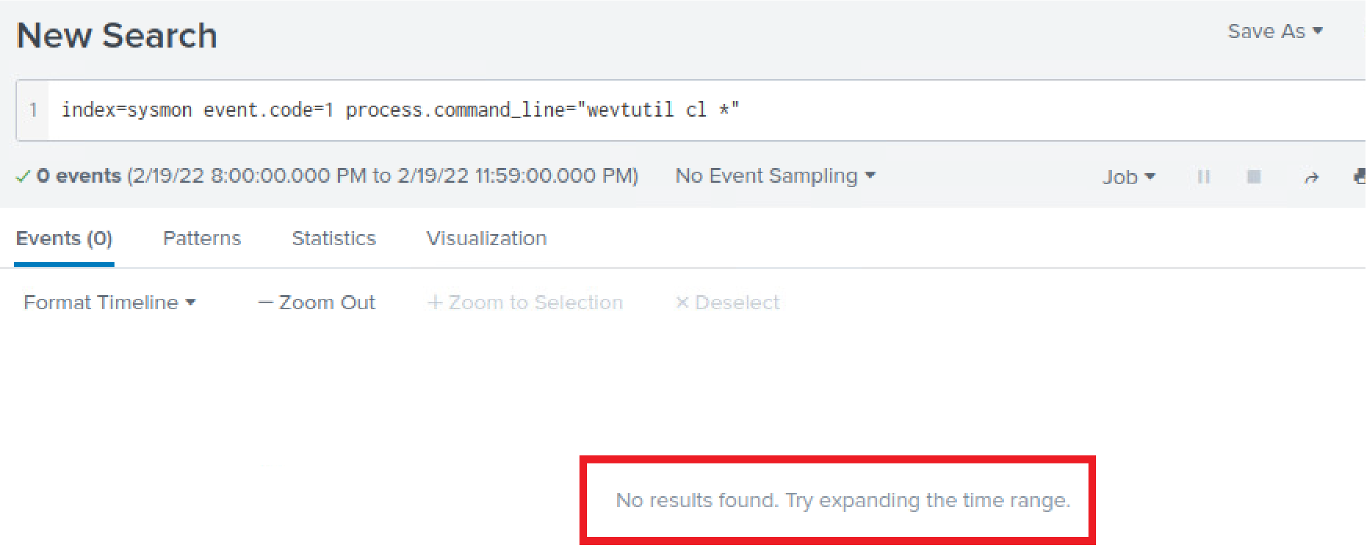
Specific Search Queries:

We will run a Splunk search in the `sysmon` index looking for `event.code` 1, and `process.command_line` containing `"wevtutil cl"`

Set date/time between 02/19/22 17:00 to 02/19/22 23:59:

Splunk Search:

```
index=sysmon event.code=1 process.command_line="wevtutil cl *"
```



```
index=sysmon event.code=1 process.command_line="wevtutil cl *"
```

✓ 0 events (2/19/22 8:00:00.000 PM to 2/19/22 11:59:00.000 PM)

No results found. Try expanding the time range.

Nothing! Bummer. OK.

FOR THE NOTES: *As the Threat Hunter, we would add this Splunk query (including the date/time specified for the query), as well as some details about the results to our notes and continue.*

So, again, are we dead in the water?
Is there nothing else we can do to find what commands the attacker used to clear the logs?

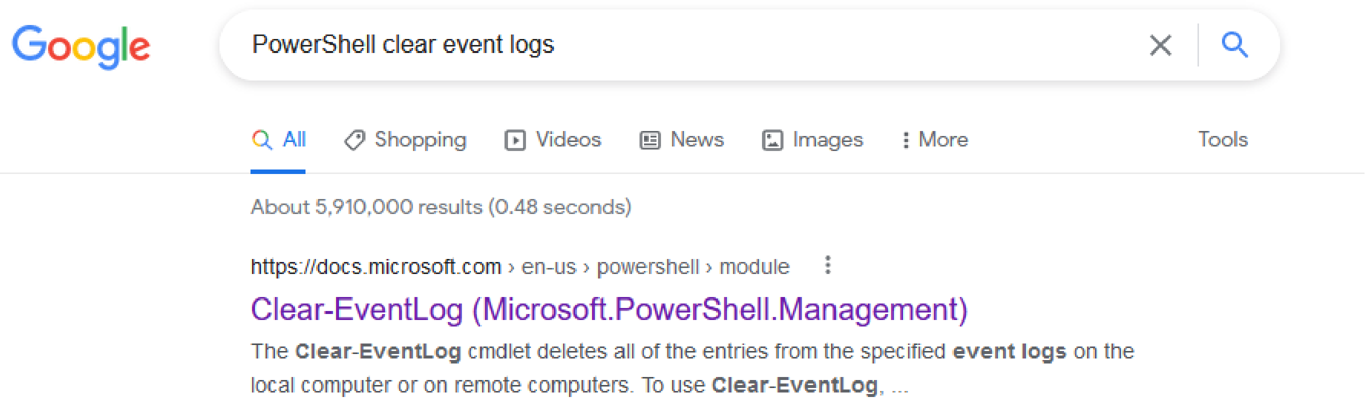
Checking our **NOTES** that we took, recall that we made note in our specific hypothesis that attackers have also been know to use **PowerShell.**

Let's move on to **PowerShell**.

Question:

How could we find a way to clear the Windows Event Logs using PowerShell?

A simple Google search for `PowerShell clear event logs` results in a link to a Microsoft article about a "Clear-EventLog" cmdlet.



PowerShell clear event logs

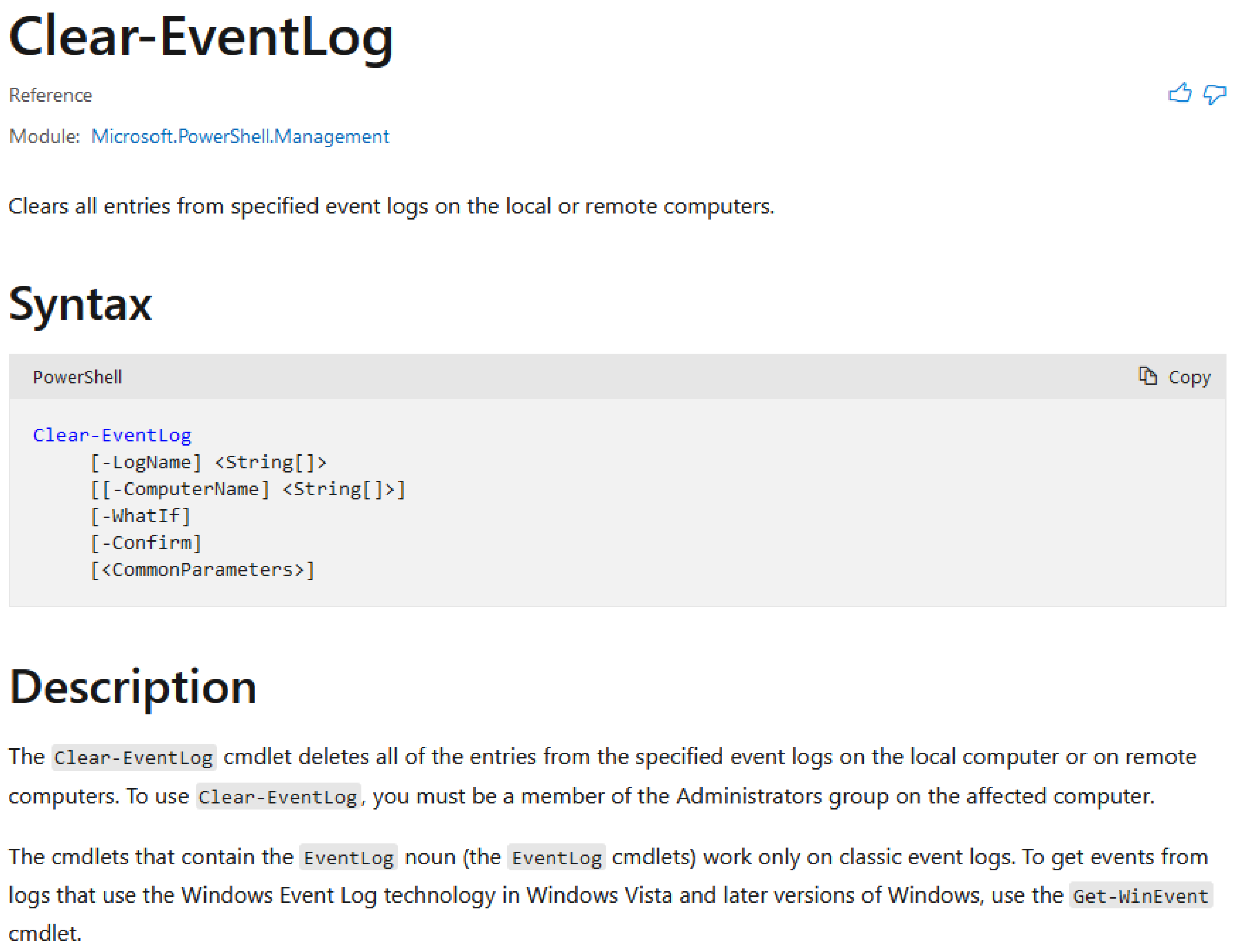
About 5,910,000 results (0.48 seconds)

https://docs.microsoft.com › en-us › powershell › module

Clear-EventLog (Microsoft.PowerShell.Management)

The **Clear-EventLog** cmdlet deletes all of the entries from the specified **event logs** on the local computer or on remote computers. To use **Clear-EventLog**, ...

Let's take a look at this article.



Clear-EventLog

Reference

Module: Microsoft.PowerShell.Management

Clears all entries from specified event logs on the local or remote computers.

Syntax

PowerShell

```powershell
Clear-EventLog
     [-LogName] <String[]>
     [[-ComputerName] <String[]>]
     [-WhatIf]
     [-Confirm]
     [<CommonParameters>]
```

Description

The `Clear-EventLog` cmdlet deletes all of the entries from the specified event logs on the local computer or on remote computers. To use `Clear-EventLog`, you must be a member of the Administrators group on the affected computer.

The cmdlets that contain the `EventLog` noun (the `EventLog` cmdlets) work only on classic event logs. To get events from logs that use the Windows Event Log technology in Windows Vista and later versions of Windows, use the `Get-WinEvent` cmdlet.

Source: https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/clear-eventlog?view=powershell-5.1

FOR THE NOTES: *As the Threat Hunter, we would add this information we have discovered about PowerShell cmdlet 'Clear-EventLog' (including any Source URLs) to our notes and continue.*

Interesting!

So, could an attacker use the PowerShell cmdlet 'Clear-EventLog' as a method to clear the logs?

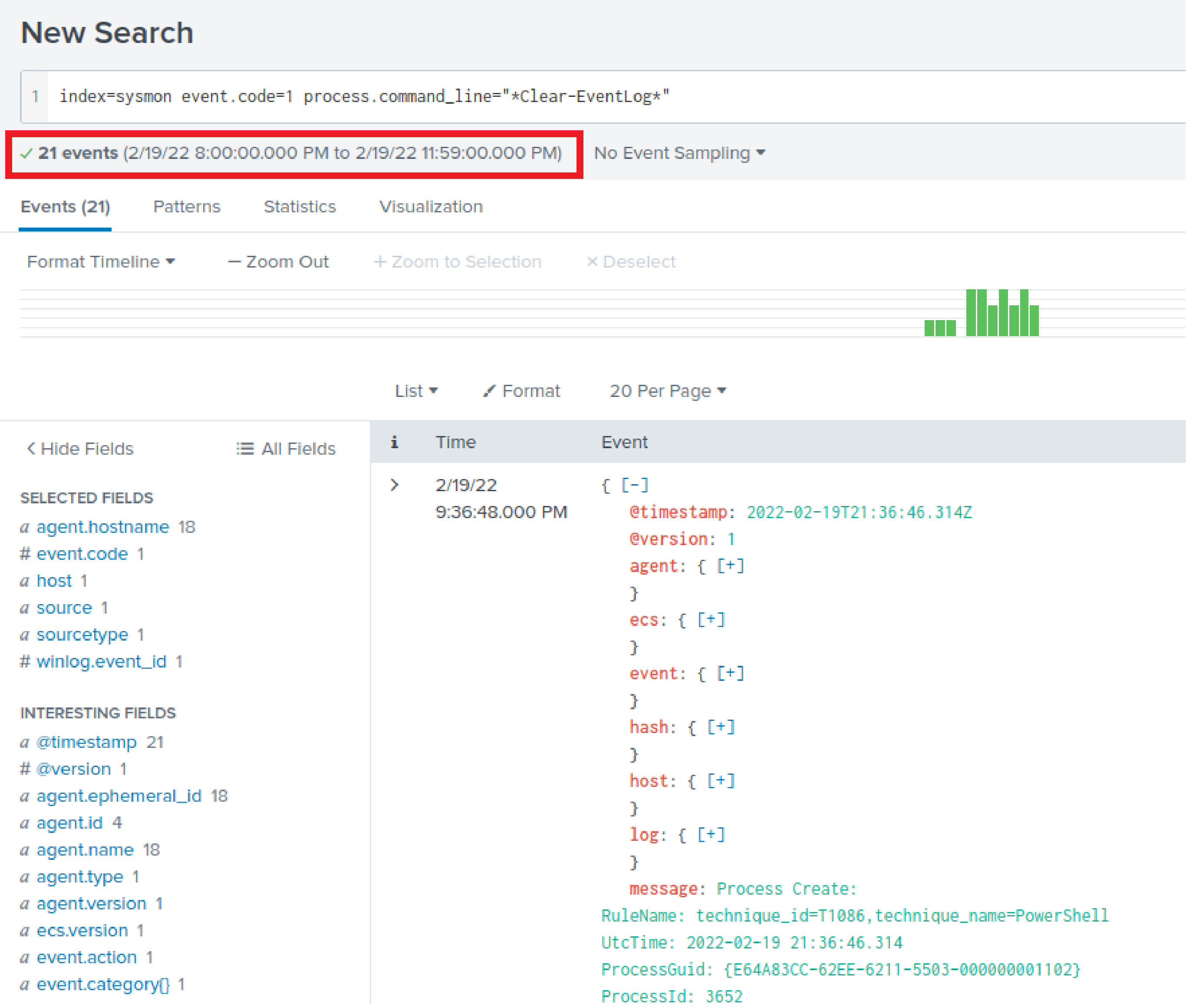
Sure! Let's dive into Splunk and hunt for any evidence that the 'Clear-EventLog' cmdlet may have been used.

Let's search for this PowerShell command in Splunk. We will search the `sysmon` index for `event.code` 1, and `process.command_line` containing `"Clear-EventLog"`

Set date/time between 02/19/22 17:00 to 02/19/22 23:59:

Splunk Search:

```
index=sysmon event.code=1 process.command_line="*Clear-EventLog*"
```



```
index=sysmon event.code=1 process.command_line="*Clear-EventLog*"
```

✓ 21 events (2/19/22 8:00:00.000 PM to 2/19/22 11:59:00.000 PM)

SELECTED FIELDS
- a agent.hostname 18
- # event.code 1
- a host 1
- a source 1
- a sourcetype 1
- # winlog.event_id 1

INTERESTING FIELDS
- a @timestamp 21
- # @version 1
- a agent.ephemeral_id 18
- a agent.id 4
- a agent.name 18
- a agent.type 1
- a agent.version 1
- a ecs.version 1
- a event.action 1
- a event.category{} 1

2/19/22 9:36:48.000 PM

```
{ [-]
   @timestamp: 2022-02-19T21:36:46.314Z
   @version: 1
   agent: { [+]
   }
   ecs: { [+]
   }
   event: { [+]
   }
   hash: { [+]
   }
   host: { [+]
   }
   log: { [+]
   }
   message: Process Create:
RuleName: technique_id=T1086,technique_name=PowerShell
UtcTime: 2022-02-19 21:36:46.314
ProcessGuid: {E64A83CC-62EE-6211-5503-000000001102}
ProcessId: 3652
```

21 Events!

Now we have proof that the attacker used the PowerShell cmdlet 'Clear-EventLog' as a method to wipe the Windows Event logs!

FOR THE NOTES: *As the Threat Hunter, we would add this Splunk query (including the date/time specified for the query), as well as some details about the results to our notes and continue.*

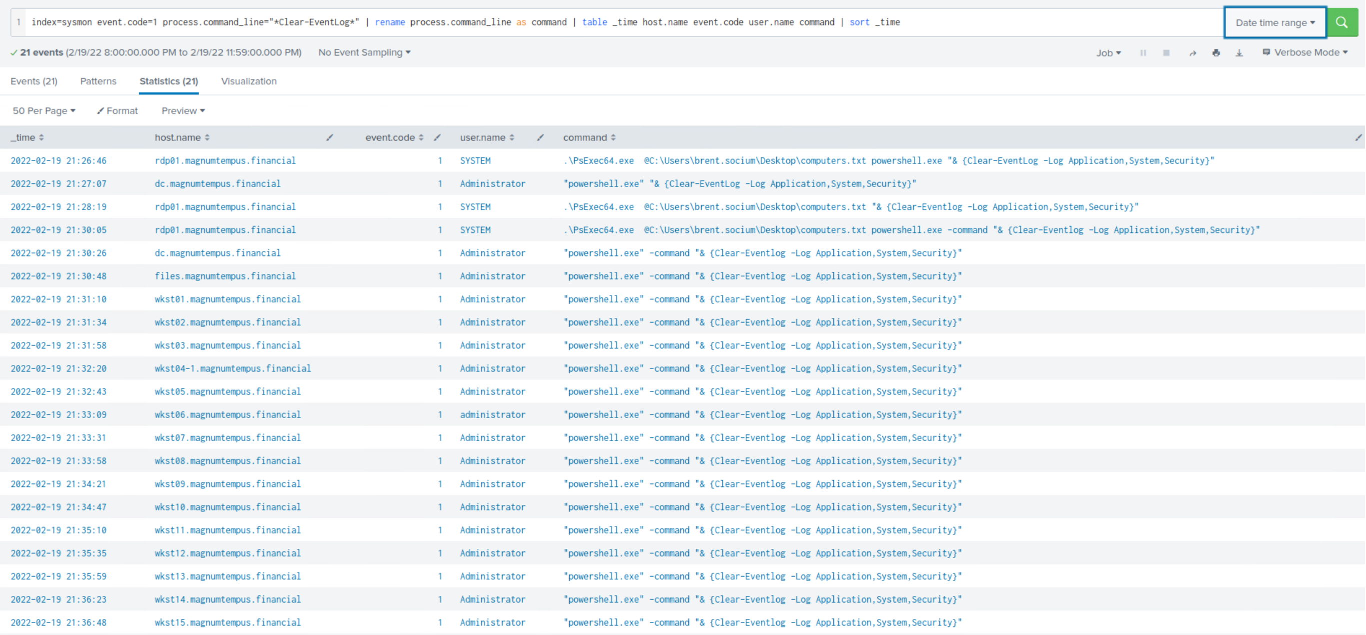
Now let's make that Splunk query more presentable.

We will search the `sysmon` index for `event.code` 1, and `process.command_line` containing `"Clear-EventLog"` AND we will use `rename` to rename the long `process.command_line` field as `command` then we will organise the results in a nice table using `table` specifying the fields we want (`"_time host.name event.code user.name command"`), and finally we will `sort` by `_time` (oldest on top to newest on the bottom). Also, note that I changed the number of events to show as 50.

Set date/time between 02/19/22 17:00 to 02/19/22 23:59:

Splunk Search:

```
index=sysmon event.code=1 process.command_line="*Clear-EventLog*" | rename process.command_line as command | table _time host.name event.code user.name command | sort _time
```



```
1  index=sysmon event.code=1 process.command_line="*Clear-EventLog*" | rename process.command_line as command | table _time host.name event.code user.name command | sort _time
```

✓ **21 events** (2/19/22 8:00:00.000 PM to 2/19/22 11:59:00.000 PM)   No Event Sampling ▾

_time	host.name	event.code	user.name	command
2022-02-19 21:26:46	rdp01.magnumtempus.financial	1	SYSTEM	.\PsExec64.exe @C:\Users\brent.socium\Desktop\computers.txt powershell.exe "& {Clear-EventLog -Log Application,System,Security}"
2022-02-19 21:27:07	dc.magnumtempus.financial	1	Administrator	"powershell.exe" "& {Clear-EventLog -Log Application,System,Security}"
2022-02-19 21:28:19	rdp01.magnumtempus.financial	1	SYSTEM	.\PsExec64.exe @C:\Users\brent.socium\Desktop\computers.txt "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:30:05	rdp01.magnumtempus.financial	1	SYSTEM	.\PsExec64.exe @C:\Users\brent.socium\Desktop\computers.txt powershell.exe -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:30:26	dc.magnumtempus.financial	1	Administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:30:48	files.magnumtempus.financial	1	Administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:31:10	wkst01.magnumtempus.financial	1	Administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:31:34	wkst02.magnumtempus.financial	1	Administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:31:58	wkst03.magnumtempus.financial	1	Administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:32:20	wkst04-1.magnumtempus.financial	1	Administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:32:43	wkst05.magnumtempus.financial	1	Administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:33:09	wkst06.magnumtempus.financial	1	administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:33:31	wkst07.magnumtempus.financial	1	Administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:33:58	wkst08.magnumtempus.financial	1	Administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:34:21	wkst09.magnumtempus.financial	1	Administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:34:47	wkst10.magnumtempus.financial	1	Administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:35:10	wkst11.magnumtempus.financial	1	Administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:35:35	wkst12.magnumtempus.financial	1	Administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:35:59	wkst13.magnumtempus.financial	1	Administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:36:23	wkst14.magnumtempus.financial	1	Administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"
2022-02-19 21:36:48	wkst15.magnumtempus.financial	1	Administrator	"powershell.exe" -command "& {Clear-Eventlog -Log Application,System,Security}"

FOR THE NOTES: *As the Threat Hunter, we would add this specific Splunk query (including the date/time specified for the query), as well as some details about the results to our notes and continue.*

How can we document the final results of the hunt?

Now, perhaps you will see how important it was to take copious notes throughout the Threat Hunting process.

We can document the final results of the hunt by using our **NOTES** to fill in the details of a Threat Hunting Template, and this process can be repeated using new hypotheses.

For this Threat Hunting exercise that we have completed, it would make sense to break these results into three separate Threat Hunting Templates:

- "Windows Event Logs Cleared (via Event Viewer GUI)"
- "Windows Event Logs Cleared via Windows Command (wevtutil cl)"
- "Windows Event Logs Cleared via PowerShell (Clear-EventLog)"

THREAT HUNTER TEMPLATES

Title: "Windows Event Logs Cleared (via Event Viewer GUI)"

Date Created: 2022-07-10

Hypothesis: Attackers will try to cover their tracks using the Event Viewer GUI

Mitre Tactic: T1070 "Indicator Removal on Host"

Mitre Sub Technique: T1070.001 "Indicator Removal on Host: Clear Windows Event Logs" (via Event Viewer GUI)

Simulation Details (if any): None

Proposed Search Query: `index=wineventlogs event.code=1102`

Hunter Limitations/Observation Notes: The Proposed Search Query did find evidence of Windows Event logs getting cleared, but the results were broad. The search query was improved to display the relevant data in a readable format (see Proposed Detection Query). It's important to note that the search results did not specifically show if the Windows Event Logs were cleared via the Event Viewer GUI, and so it would be prudent to test this in a simulation to verify that Windows Security Event ID 1102 would still be triggered (no matter what method an attacker used: GUI, Command, or PowerShell).

Hunt Findings: Windows Event 1102 is wonderful for detecting when Windows Event logs are cleared, and the query developed below should be made into a detection.

Proposed Detection Title: "ALERT: Windows Event 1102 - The audit log was cleared"

Proposed Detection Query:

```
index=wineventlogs event.code=1102 | rename winlog.user_data.SubjectUserName as user | table _time host.name event.code winlog.task user | sort _time
```

Title: "Windows Event Logs Cleared via Windows Command (wevtutil cl)"

Date Created: 2022-07-10

Hypothesis: Attackers will try to cover their tracks using Windows Commands (such as "wevtutil cl")

Mitre Tactic: T1070 "Indicator Removal on Host"

Mitre Sub Technique: T1070.001 "Indicator Removal on Host: Clear Windows Event Logs" (via Windows Command "wevtutil cl")

Simulation Details (if any): None

Proposed Search Query: `index=sysmon event.code=1 process.command_line="wevtutil cl *"`

Hunter Limitations/Observation Notes: The Proposed Search Query did not produce any valuable results. However, it would be prudent to run a simulation to test this command, and build the detection.

Hunt Findings: Although the query did not produce results, I believe the developed query should be made into a detection because attackers could possibly use the command to clear the Windows Event logs.

Proposed Detection Title: "ALERT: Windows Command ("wevtutil cl") used to clear Windows Event logs"

Proposed Detection Query:

```
index=sysmon event.code=1 process.command_line="wevtutil cl *" | rename process.command_line as command | table _time host.name event.code user.name command | sort _time
```

Title: "Windows Event Logs Cleared via PowerShell (Clear-EventLog)"

Date Created: 2022-07-10

Hypothesis: Attackers will try to cover their tracks using PowerShell

Mitre Tactic: T1070 "Indicator Removal on Host"

Mitre Sub Technique: T1070.001 "Indicator Removal on Host: Clear Windows Event Logs" (via PowerShell Clear-EventLog)

Simulation Details (if any): None

Proposed Search Query: `index=sysmon event.code=1 process.command_line="*Clear-EventLog*"`

Hunter Limitations/Observation Notes: The Proposed Search Query produced evidence that the PowerShell cmdlet "Clear-EventLog" was used, but the results were broad. The search query was improved to display the relevant data in a readable format (see Proposed Detection Query).

Hunt Findings: Use of the PowerShell cmdlet 'Clear-EventLog' should be monitored for any suspicious behavior, and so the Proposed Detection Query should be made into a detection.

Proposed Detection Title: "ALERT: PowerShell 'Clear-EventLog' executed"

Proposed Detection Query:

```
index=sysmon event.code=1 process.command_line="*Clear-EventLog*" | rename process.command_line as command | table _time host.name event.code user.name command | sort _time
```