FourCore Article

Last Updated On **Mon Jun 06, 2022**

Using Windows Event Log IDs For Threat Hunting

Written By **Swapnil**
Co-Founder @ FourCore

Workflow to setup detections

Windows Event Logs

In Cybersecurity, a new threat emerges every day, and it is essential to set up detections and logging to detect if any of these threats are attacking your organization. Furthermore, when an adversary compromises your organization, it is vital to stop them as soon as possible. Therefore, it is essential to have updated detection strategies and capabilities to identify these threats in real-time and stop them in their tracks. One of the best resources for detecting security-related events in Windows is the Windows Event Log.

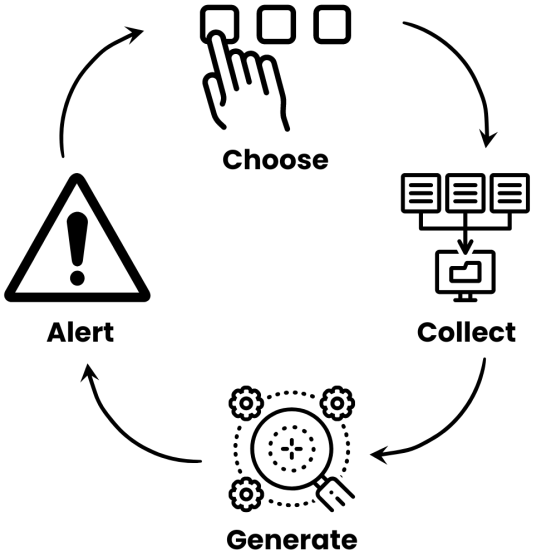
Windows Event Log is a tool present in Windows which keeps a detailed record of

System, Security and Application Logs. Administrators can use these logs to troubleshoot issues, find errors, and ensure correct security configurations are followed. In addition, each event in Windows Event Logs also matches a specific Event ID, which tools can use to notify and detect particular actions. Security Events in Windows Event Logs provide a wealth of data that can detect an adversary or be used during forensic analysis of the compromised system.

Detections using Event Logs

Security Events store information based on the system's audit policies. For example, the security log can be configured to log an entry when a user logs in. In addition, security event logs contain the date, time, user, computer, source, and type of event. These events can also be consumed or forwarded to various SIEM/logging solutions, which Security teams can then use to build detections and alert on specified events and threshold of events.

Windows Event IDs have around 85% coverage of Windows Specific techniques in MITRE ATT&CK. Therefore, a workflow can be established in the organization to cover these techniques and ensure that you have set up proper detection and alerting.



Workflow to setup detections

- **Choose**: Select a necessary technique depending upon the latest threat intelligence or something essential in your organization.
- **Collect**: Collect all the necessary data on the technique, such as log sources,

event IDs, descriptions etc.

- **Generate**: Generate logs for that event using tools or manually performing the action. Ensure that the records are being ingested.
- **Alert**: Create an optimized query to detect the potential threat. Ensure low false positives.

Once this methodology is perfected, you can become proactive and start threat hunting in your environment. Threat hunting is becoming increasingly important if you need to stay ahead of the latest cyber threats.

Optimizing Detections

When enabled, Windows will provide lots of logs for various events on the system. But, more than likely, they would add noise to your logging infrastructure. Therefore, it is crucial to disable the collection of logs which are highly likely to be false positives and identify other means or sources to collect more enriched data. Some examples of important logs which should be enabled for detection are given below. It is up to the organization to baseline its requirements and allow event logs while reducing false positives.

- **Event ID 4688** - A new process has been created: Event 4688 documents each program that is executed, who the program ran as and the process that started this process.

  When you start a program you are creating a "process" that stays open until the program exits. This process is identified by the Process ID:. You can correlate this event to other events by Process ID to determine what the program did while it ran and when it exited

```
A new process has been created.

Creator Subject:
 Security ID:  SYSTEM
 Account Name:  RFSH$
 Account Domain:  LAB
 Logon ID:  0x3E7

Target Subject:
```

```
 Security ID:  LAB\rsmith
 Account Name:  rsmith
 Account Domain:  LAB
 Logon ID:  0x2C9D82

Process Information:
 New Process ID:  0x2e0e4
 New Process Name: C:\Windows\System32\RuntimeBroker.exe
 Token Elevation Type: %%1938
 Mandatory Label:  Mandatory Label\Medium Mandatory Level
 Creator Process ID: 0x268
 Creator Process Name: C:\Windows\System32\svchost.exe
 Process Command Line:
```

- **Event ID 4720** - A user account was created: When a new user account is made in a windows workstation, there would be an event log with ID 4720. Since a majority of accounts are created in Active Directory, this could be an indicator of an attempt of persistence.

```
A user account was created.

Subject:

   Security ID:  ACME-FR\administrator
   Account Name:  administrator
   Account Domain:  ACME-FR
   Logon ID:  0x20f9d

New Account:

   Security ID:  ACME-FR\John.Locke
   Account Name:  John.Locke
   Account Domain:  ACME-FR

Attributes:

   SAM Account Name: John.Locke
   Display Name:  John Locke
   User Principal Name: John.Locke@acme-fr.local
   Home Directory:  -
   Home Drive:  -
   Script Path:  -
```

```
    Profile Path:  -
    User Workstations: -
    Password Last Set: <never>
    Account Expires:  <never>
    Primary Group ID: 513
    Allowed To Delegate To: -
```

- **Event ID 1102** - The audit log was cleared: Event 1102 is logged whenever the Security log is cleared, REGARDLESS of the status of the Audit System Events audit policy. The Account Name and Domain Name fields identify the user who cleared the log.

```
The audit log was cleared.
Subject:
Security ID: WIN-R9H529RIO4Y\Administrator
Account Name: Administrator
Domain Name: WIN-R9H529RIO4Y
Logon ID: 0x169e9
```

- **Event ID 4798** - A user's local group membership was enumerated: Windows logs this event when a process enumerates the local groups to which a the specified user belongs on that computer. This event is valuable for catching so-called APT actors who are scoping out the local accounts on a system they have compromised so that they extend their horizontal kill chain.

```
A user's local group membership was enumerated.

Subject:
  Security ID: AzureAD\RandyFranklinSmith
  Account Name: RandyFranklinSmith
  Account Domain: AzureAD
  Logon ID: 0x7A1EA

User:
   Security ID: DESKTOP-TMO9MI9\Administrator
   Account Name: Administrator
   Account Domain: DESKTOP-TMO9MI9

 Process Information:
```

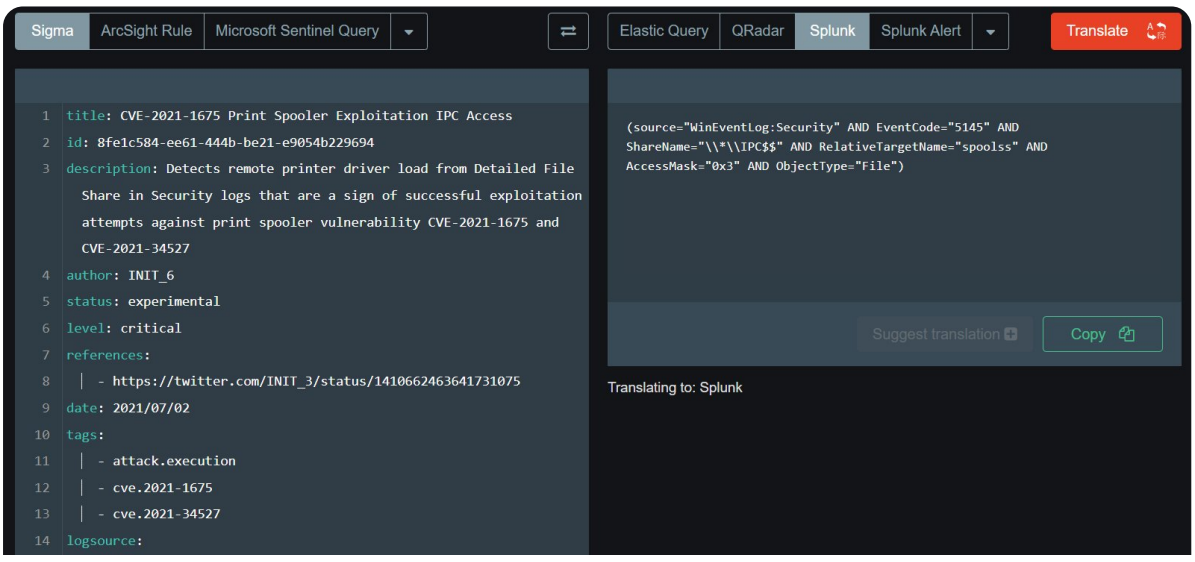
```
    Process ID: 0x106c
    Process Name: C:\Windows\System32\mmc.exe
```

Similarly, Windows keeps a log of many security events and can be turned on or off depending upon your needs and compliance policies. You can find a reference to these event IDs in the Windows Security Encyclopedia.

Detection Engineering with Sigma Rules

Collecting events from the Windows Event Log is not enough; it is essential to write correlation rules to detect these activities. Each security solution will have a different syntax to implement detection and alert rules. Writing new rules can become cumbersome when multiple detection systems are in use. Sigma by SigmaHQ is a generic and open YAML-based signature format that that enables a security operations team to describe relevant log events in a flexible and standardized format. Sigma rules can be converted to queries specific to your security solution. Sigma also provides an open-source list of rules you can use readily.

Taking an example from the CVE-2021-1675 Print Spooler Vulnerability, we can see how we can quickly search for an event based on EventID 5145 and filter on object type and access mask. Ability to quickly search for exploitation attempts or detect them while happening allows you to respond in a swift manner to contain a host and prevent breaches across the complete network.



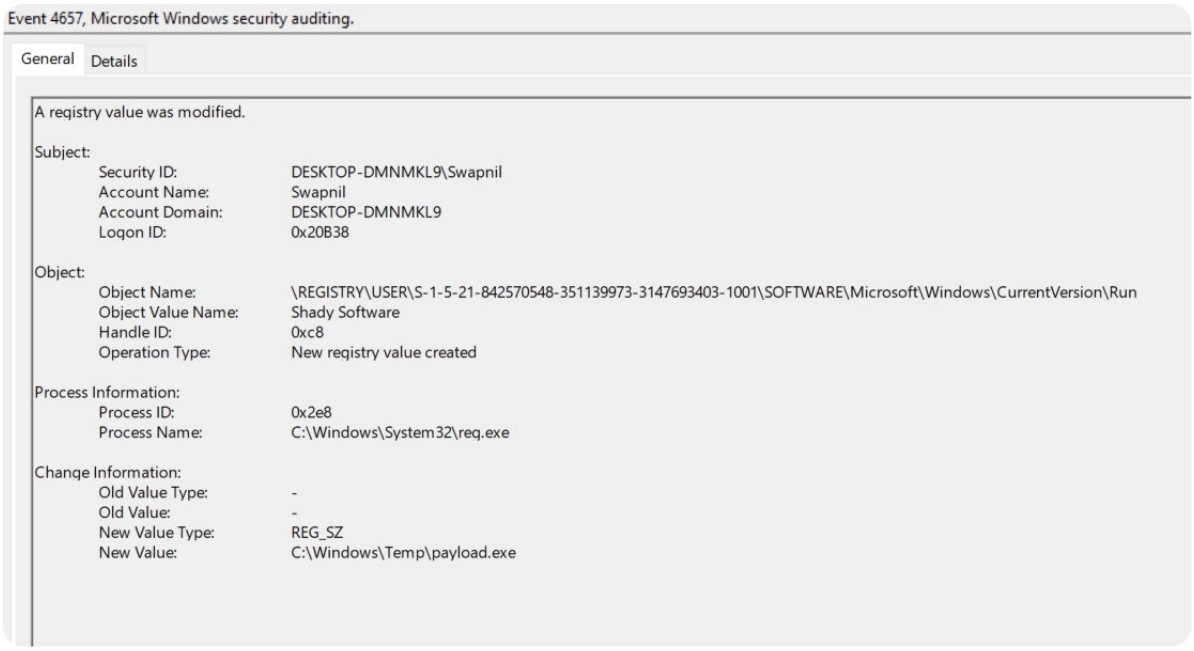
```
    product: windows
    service: security
detection:
    selection:
        EventID: '5145'
        ShareName: '\\*\IPC$'
        RelativeTargetName: 'spoolss'
        AccessMask: '0x3'
        ObjectType: 'File'
    condition: selection
```

Print Spooler Exploitation Detection

Simulate a threat yourself

Let's take the example of **Registry Run Key Persistence Technique**.

MITRE ATT&CK mentions *Registry Run Keys* as "Adversaries may achieve persistence by adding a program to a startup folder or referencing it with a Registry run key. Adding an entry to the "run keys" in the Registry or startup folder will cause the program referenced to be executed when a user logs in. These programs will be executed under the context of the user and will have the account's associated permissions level."



Registry Run Key Modification Event Log

We can create events for this technique using the firedrill utility. You can find the source here and customize it to run more such techniques to improve your detection capabilities. This technique uses the Registry Key `HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` to execute a sample payload. Running this technique will add a value which would be captured in the Windows Event Logs with event ID 4657. An example Sigma Rule by Florian Roth to detect this

attack. A huge thanks to Florian and Nextron Systems for maintaining and adding new rules to the Sigma repository.

```yaml
title: Reg Add RUN Key
id: de587dce-915e-4218-aac4-835ca6af6f70
description: Detects suspicious command line reg.exe tool adding key to RUN
status: experimental
date: 2021/06/28
author: Florian Roth
references:
  - https://app.any.run/tasks/9c0f37bc-867a-4314-b685-e101566766d7/
  - https://docs.microsoft.com/en-us/windows/win32/setupapi/run-and-runonce-
logsource:
  category: process_creation
  product: windows
detection:
  selection:
    CommandLine|contains|all:
      - "reg"
      - " ADD "
      - 'Software\Microsoft\Windows\CurrentVersion\Run'
  condition: selection
falsepositives:
  - Unknown
level: medium
tags:
  - attack.persistence
  - attack.t1547.001
```

Conclusion

It is crucial to stay on top of emerging threats and contain or detect them in real-time. In a Security Operations Center, collecting Security Logs from Windows Event Logs and using them is essential. It is imperative to set up these detections and baseline the events in your organization to detect these threats swiftly. The baselines in your organization could be created manually or by using some automated tool to simplify the process.

Threat Hunting with FourCore ATTACK

You can validate your detection rules and alerts with FourCore ATTACK. Optimize your rules by simulating attackers' behaviour generating different Event IDs you can utilize for validating detections! Get a free assessment with a free trial of FourCore ATTACK.

References

- Sigma and Sigma Rules
- Windows Security EventID Reference
- Sigma Convertor
- Splunk SIEM

Read More

FAQ

What is BAS?

Breach and Attack Simulation

Automated Penetration Testing

Continuous Automated Red Teaming

Twitter GitHub LinkedIn NCOE-DSCI