Sysmon 2.0 & EventViz

By Russ McRee – ISSA Senior Member, Puget Sound (Seattle) Chapter

Prerequisites

Windows operating system

R 3.1.2[1] and RStudio[2] for EventViz

Congratulations and well done to Josh Sokol for winning *2014 Toolsmith Tool of the Year* with his very popular SimpleRisk!

Sysmon 2.0 was welcomed to the world on 19 JAN 2015, warranting immediate attention as part of the state of cybersecurity focus for February's *ISSA Journal*. If you want to better understand the state of cybersecurity on your Windows systems, consider System Monitor (Sysmon) a requirement. Sysmon is "a Windows system service and device driver that, once installed on a system, remains resident across system reboots to monitor and log system activity to the Windows event log. It provides detailed information about process creations, network connections, and changes to file creation time."[3] The real upside is that you can ship related events using Windows Event Collection or the agent provided as part of your preferred SIEM implementation. You can also conduct simple exports of EVTX files during forensic or malware runtime analysis for parsing and queries. I built EventViz in R, as a Shiny[4] app, to simplify this process and read CSV exports from Windows Event Viewer. It's a work in progress to be certain, but one you can make immediate use of in order to pivot on key data in Sysmon logs.

I pinged Thomas Garnier, who along with Microsoft Azure CTO Mark Russinovich, created Sysmon. Thomas pointed out that the need to understand our networks—how they are used or abused—has never been higher. He also reminded that, unfortunately, there is a gap between the information needed by network defenders and the information provided by operating systems across versions. To that end, "Sysinternals Sysmon was created to provide rich information across OS versions while running in the background, staying resident across reboots. It provides detailed information on process creation, image loading, driver loading, network connections, and more. It allows you to easily filter generated events and update its configuration while it is still running. All the activities are captured in the Windows event log to integrate with existing Windows Event Collection or SIEM solutions."

Of the eight Event IDs generated by Sysmon, you can consider six immediately useful for enhancing situational awareness and strengthened defenses. You'll want to tune and optimize how you configure Sysmon so you don't flood your logging systems with data you determine later isn't as helpful as you hoped. The resulting events can be quite noisy given the plethora of data made available per the following quick event overview:

Event ID 1: Process creation
The process creation event provides extended information about a newly created process.

Event ID 2: A process changed a file creation time
The change file creation time event is registered when a file creation time is explicitly modified by a process. Attackers may change the file creation time of a backdoor to make it look like it was installed with the operating system, but many processes legitimately change the creation time of a file.

Event ID 3: Network connection
The network connection event logs TCP/UDP connections on the machine. It is disabled by default.

Event ID 4: Sysmon service state changed
The service state change event reports the state of the Sysmon service (started or stopped).

Event ID 5: Process terminated
The process terminate event reports when a process terminates.

Event ID 6: Driver loaded
The driver loaded events provides information about a driver being loaded on the system.

Event ID 7: Image loaded
The image loaded event logs when a module is loaded in a specific process. This event is disabled by default and needs to be configured with the –l option.

I love enabling the monitoring of images loaded during malware and forensic analysis, but as the Sysmon content says, "this event should be configured carefully, as monitoring all image load events will generate a large number of events." I'll show you these Event IDs come to play during review of a system compromised by a Trojan:Win32/Beaugrit.gen!AAA[5]

1 http://mran.revolutionanalytics.com/download/.
2 http://www.rstudio.com/products/RStudio/.
3 https://technet.microsoft.com/en-us/sysinternals/dn798348.
4 http://shiny.rstudio.com/.
5 https://www.virustotal.com/en/file/8a4e23ab8f295baec33700fff23f03611037c7ef15d54f85a1a745d44144f254/analysis/.

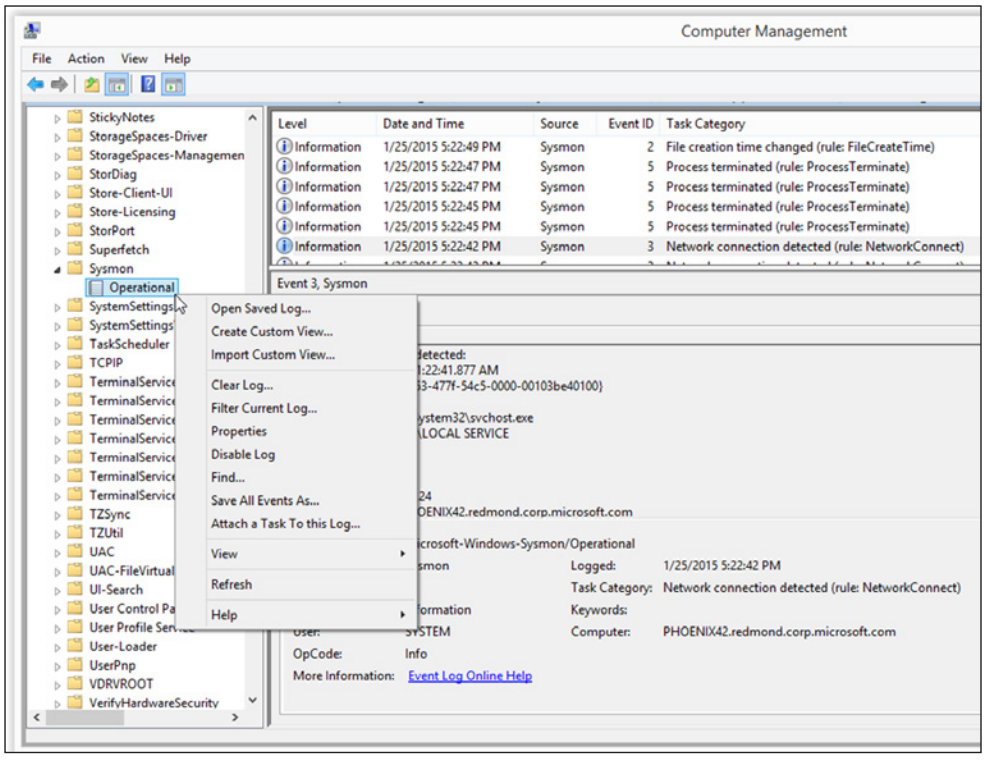

Figure 1 – Sysmon operational log entries in event viewer

sample using EventViz to analyze Sysmon logs. Beaugrit.gen is a rootkit that makes outbound connections to request data and download files while also interacting with Internet Explorer.

Sysmon installation

I had the best luck downloading Sysmon and unpacking it into a temp directory. From an administrator command prompt I changed directory to the temp directory and first ran `Sysmon.exe -accepteula –i`. This accepts the EULA, installs Sysmon as a service, and drops Sysmon.exe in C:\Windows, making it available at any command prompt path. I exited the first administrator command prompt, spawned another one, and ran `sysmon –m`. This step installs the event manifest, which helps you avoid verbose and erroneous log messages such as "The description for Event ID 5 from source Microsoft-Windows-Sysmon cannot be found." I followed that with `sysmon -c -n -l -h md5,sha1,sha256`. The `-c` flag updates the existing configuration, `-n` enables logging of network connections (Event ID 3), `-l` enables logging the loading (can be noisy) of modules (Event ID 7), and `-h` defines what hashes you wish to collect as part of event messaging. You may be happier with just one hash type configured to again reduce volume. Once installed and properly configured you'll find Sysmon logs in the Event Viewer under *Applications and Services Logs => Microsoft => Windows => Sysmon => Operational* as seen in figure 1.

The physical system path is *%SystemRoot%\System32\Winevt\Logs\Microsoft-Windows-Sysmon%4Operational.evtx*. You can optionally define configuration preferences with a config.xml file; refer to Sysmon documentation for specifics. To create Sysmon CSV files that can be analyzed with EventViz, right click the Operational log as seen in figure 1 and Save All Event As sysmon.csv. The will result in a CSV version of all Sysmon log entries written to the system you are analyzing. Remember that all Windows event logs are set to 65536KB and to overwrite as needed by default. You'll likely want to factor for updating this to ensure an appropriately sized log sample in order to conduct proper investigations, particularly if you're not shipping the logs off system.

Analysis with EventViz

Collecting logs is one thing, but making quick use of them is the real trick. There are so many ways and means with which to review and respond to particular events as defined by detection and rules logic. Piping Sysmon logs to your collection mechanism is highly recommended. Windows Event Collection/Windows Event Forwarding are incredibly useful, the results of which can be consumed by numerous commercial products. Additionally there are free and open source frameworks that you can leverage. Enterprise Log Search and Archive[6] (ELSA) immediately comes to mind as does the ELK stack (Elasticsearch, Logstash, Kibana). See Josh Lewis' "Advanced Threat Detection with Sysmon, WEF and Elasticsearch (AKA Panther Detect)"[7] as a great reference and pointer.

There are also excellent uses for Sysmon that don't require enterprise collection methods. You may have single instances of dedicated or virtual hosts that are utilized for runtime analysis of malware and/or other malfeasance. I utilized just such a Windows 7 SP1 virtual machine to test Sysmon capabilities with the above mentioned Beaugrit.gen sample. You can of course utilize the built-in Windows Event Viewer, but ease of use and quick pivots and queries are not its strong suit. I've started on EventViz to address this issue and plan to keep developing against it. For folks with an appreciation for R, this is a nice exemplar for its use. If you need a good primer on using R for information security-related purposes, I've got you covered there. In January, *ADMIN Magazine* published my article, "Security Data Analytics and Visualization with R," [8] which is a convenient and directly useful way for you to get your feet wet with R.

6 https://code.google.com/p/enterprise-log-search-and-archive/.

7 http://joshuadlewis.blogspot.com/2014/10/advanced-threat-detection-with-sysmon_74.html?m=1.

8 http://www.admin-magazine.com/Archive/2014/24/Security-data-analytics-and-visualization-with-R.

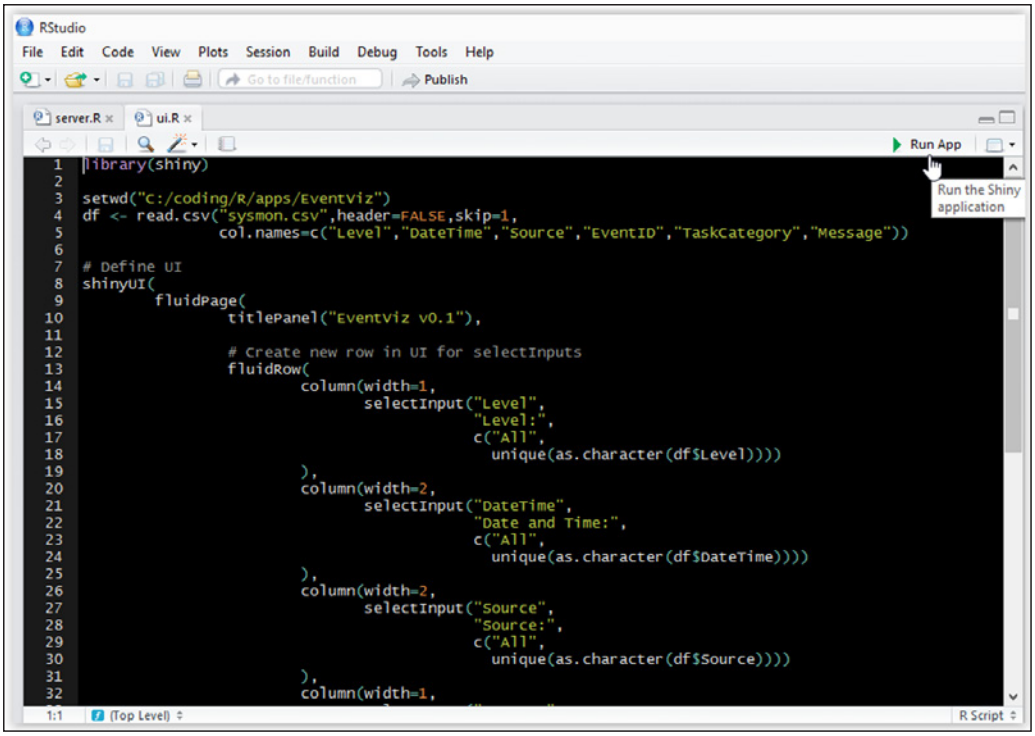

Figure 2 - Run the EventViz Shiny app

Use of EventViz currently assumes you've got a version of R installed, as well as RStudio. At an RStudio console prompt be sure to run `install.packages("shiny")` as EventViz is a Shiny app that requires the Shiny package. Create a directory where you'd plan to store R scripts,; create an apps directory therein, and an EventViz directory in the apps directory. Mine is *C:\coding\R\apps\EventViz* as an example. Copy server.R and ui.R, as well as the example CSV file we're discussing here, to the EventViz directory; you can download them from my Github EventViz repository.[9] Open server.R and ui.R and click Run App as seen in figure 2.

Give it a few minutes. It may take a bit to load as it is a crowded log set given the verbosity of Event ID 7 (Image loaded)—again, enable it with caution. I'm working on EventViz performance with larger files, but you'll see how Event ID 7 helps us here though. Once it's loaded, you'll have figure 3 in a Shiny window. You also have the option to open it in a browser.

Each of the drop-down menus represents a column heading in the sysmon.csv, albeit with a little manipulation via R where I renamed them and added a header for the messages column.

I'll work with a bit of insider knowledge, given my familiarity with the malware sample, but as long as you have a potential indicator of compromise (IOC) such as an IP address or malicious executable name, you can get started. I knew that the sample phones home to 920zl.com. When I conducted a lookup on this domain (malicious), it returned an IP address of 124.207.29.185 in Beijing. You may now imagine my shocked face. Let's start our results analysis and visualization with that IP address. I copied it to the EventViz search field and it quickly filtered two results from 9270 entries as seen in figure 4. This filter is much faster than the initial app load as the whole data set is now immediately available in memory. This is both a benefit and a curse with R. It performs well once

9 https://github.com/holisticinfosec/EventViz.

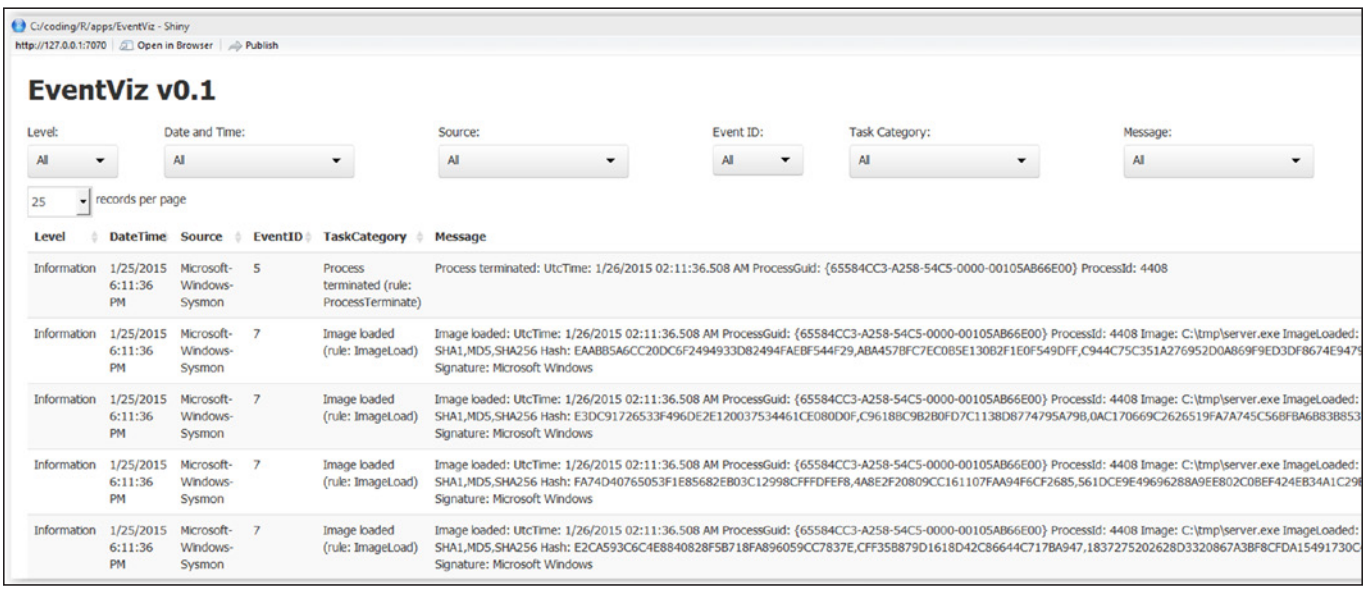

Figure 3 - EventViz UI

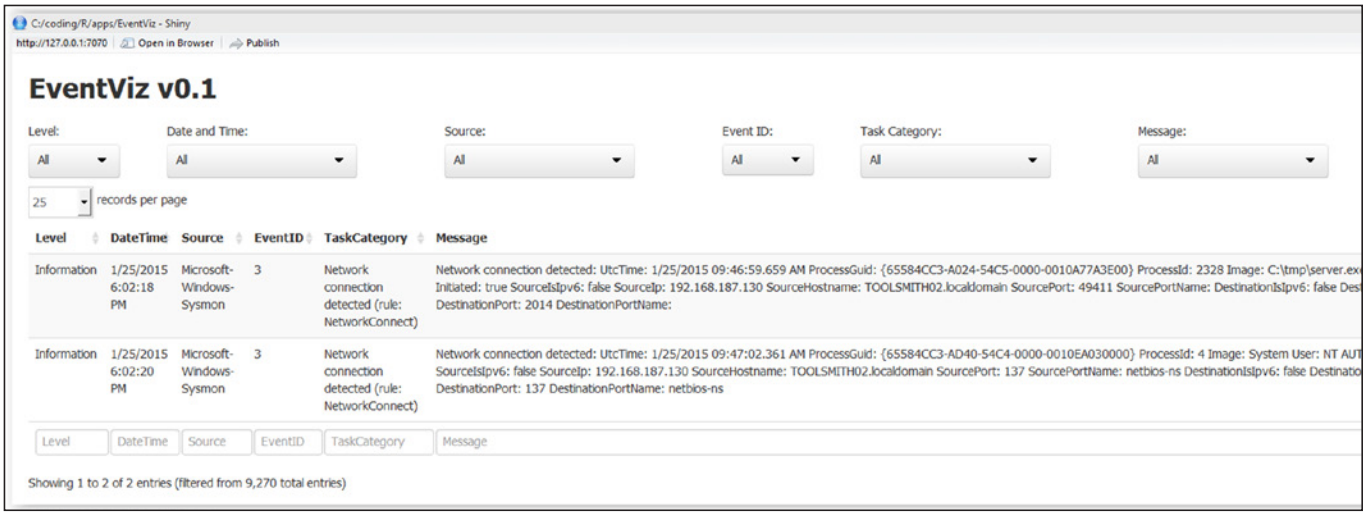

Figure 4 - EventViz IP address search results

loaded but a memory hog thereafter, and it's not well known for cleaning up after itself.

Sysmon Event ID 3, which logs detected network connections, trapped C:\tmp\server.exe making a connection to our suspect IP address. Nice, now we have new pivot options. You could use the Event ID drop-down to filter for all Event ID 3s for all network connections. I chose to filter Sysmon Event ID 7 and searched server.exe. The results directly matched Virus Total behavioral information for this sample, specifically the runtime DLLs. Sysmon's Event ID 7 ImageLoad logic clearly shows server.exe acting as indicated by VirusTotal per figure 5.

Drill in via Event ID 1 ProcessCreate and you'll find that server.exe was spawned by explorer.exe. The victim (me) clicked it (derp). There are endless filter and pivot options, given the data provided by Sysmon and quick filter capabilities in EventViz. Eventually (pun intended), EventViz will allow you to also analyze other Windows event log types such as the security log.

In conclusion

Sysmon clearly goes above and beyond default Windows event logging by offering insightful and detailed event data. Coupled with collection and SIEM deployments, Sysmon can be an incredible weapon as part of your detection logic. Marry your queries up with specific threat indicators and you may be both pleased and horrified (in a good way) with the results. Watch the TechNet blog and the Sysinternals site for further updates to Sysmon. For quick reviews during runtime analysis or dirty forensics, a viewer such as EventViz might be useful, assuming you export to your Sysmon EVTX file to CSV. Keep an eye on the Github site for improvements and updates. I plan to add a file selector so you can choose from a directory of CSV files. For other feature requests you can submit via Github.

Ping me via email or Twitter if you have questions (russ at holisticinfosec dot org or @holisticinfosec).

Cheers…until next month.

Acknowledgements

—Thomas Garnier, Sysmon 2.0 developer

About the Author

Russ McRee manages the Threat Intelligence & Engineering team for Microsoft's Online Services Security & Compliance organization. In addition to toolsmith, he's written for numerous other publications, speaks regularly at events such as DEFCON, Black Hat, and RSA, and is a SANS Internet Storm Center handler. As an advocate for a holistic approach to the practice of information assurance Russ maintains holisticinfosec.org. He serves in the Washington State Guard as the Cybersecurity Advisor to the Washington Military Department. Reach him at russ at holisticinfosec dot org or @holisticinfosec.

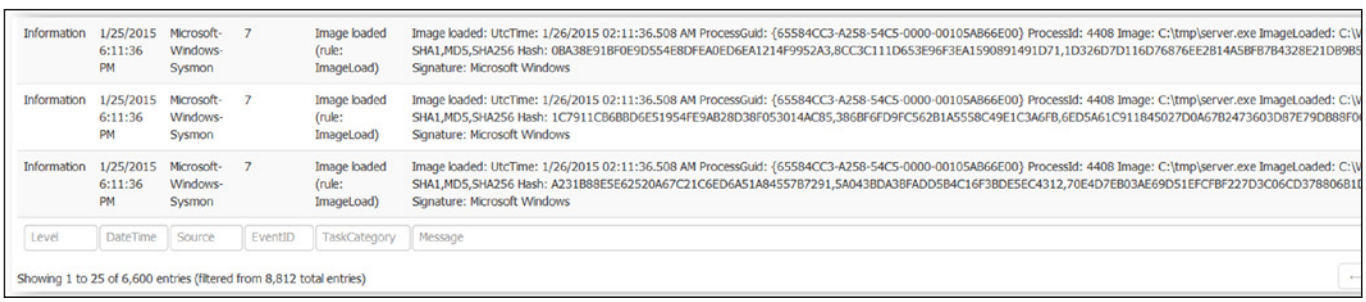

Figure 5 – EventID 7 ImageLoad matches VirusTotal