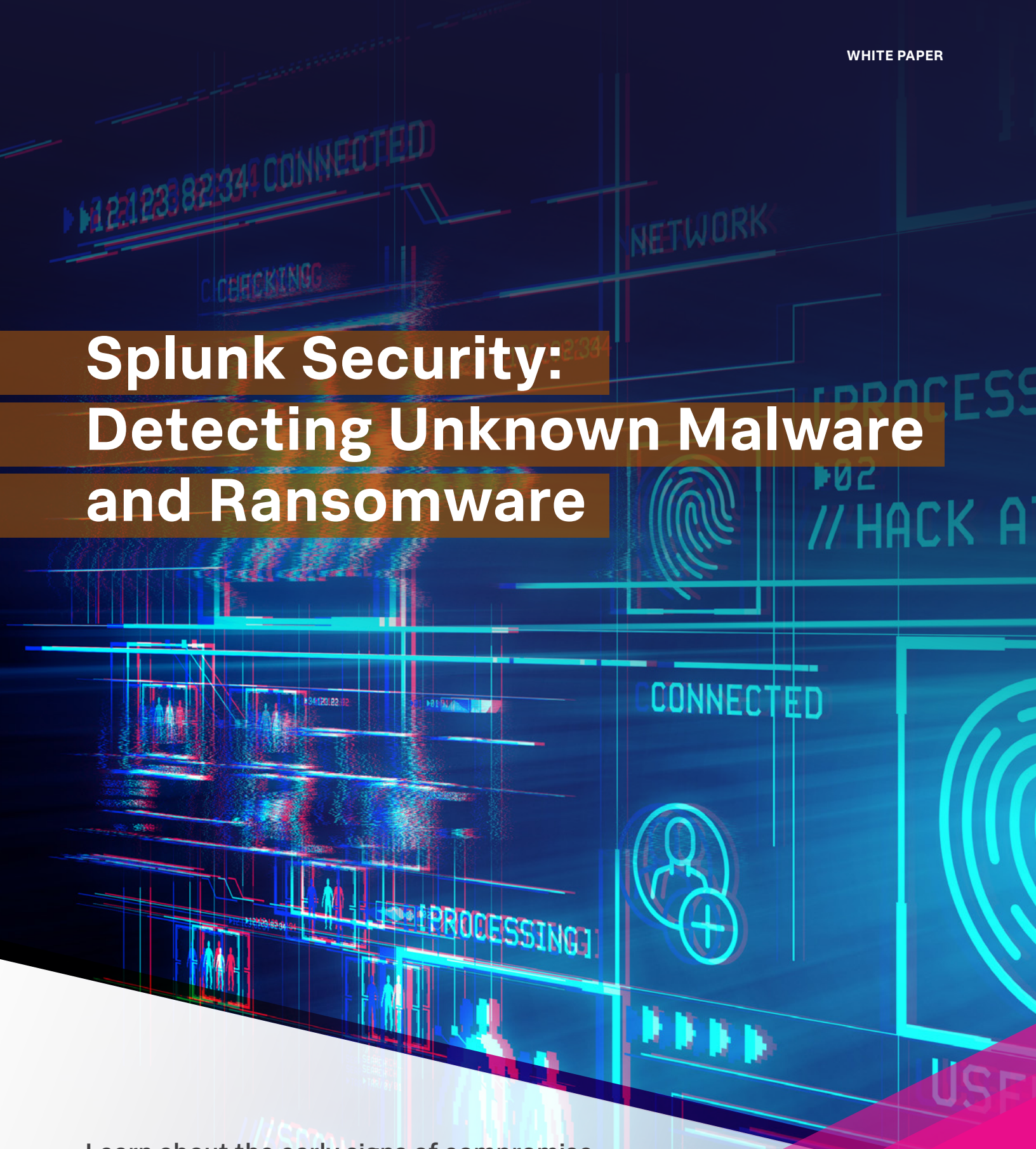
WHITE PAPER

# Splunk Security: Detecting Unknown Malware and Ransomware




Learn about the early signs of compromise with Windows Sysinternal

splunk>
turn data into doing™

Ransomware is a specific type of malware that holds data "hostage," and is especially disruptive to business due to its data-destructive nature. The ransomware threat doesn't need to keep security practitioners up at night. Detection of ransomware is key to removing compromised devices from an infected network but a holistic approach to security, centered around prevention, is necessary to keep organizations from falling victim to malware attacks.

This paper will take users on a step-by-step journey on how to detect unknown malware activity and early signs of compromise in a Windows environment. These techniques can be applied to detect malware and ransomware using Windows Systinternal events.

## The Challenge to Detecting Malware

The traditional way of detecting an advanced malware or threat compromise in a Windows environment relies on using a signature based anti-virus or malware product. But this approach can be difficult for many. Most anti-malware solutions that are signature based rely on a known list of signatures. And this comes with challenges because signature based detection will not catch everything because:

- Endpoint protection products don't have a perfect list of threats to detect all signatures that exist
- They don't apply to new types of threats that are executed as new executables at the endpoints because there is no known signature to compare against

This traditional approach is forcing organizations to deal with security breaches ranging from data exfiltration, service interruptions and ransomware that all center with the inability to protect and detect the activities on endpoints.

Fundamentally the problems lie with organizations being unable to utilize the Windows system activities events that could be collected from Windows infrastructure. As well as applying analytics to that data, to determine what is normal versus what is abnormal, by reviewing all the processes and sessions created at Windows Endpoint.

The challenges with collecting sysinternal data from all endpoint is that it requires coordinating efforts and proper outside technology that installs a light agent at Windows Endpoint that could collect granular sysinternal events in real time from many Windows systems. Once the details of the Windows activity, in event log format from the endpoint is collected, it needs to be stored in a data platform that could handle the volume of messages and be able to search and analyze system activities effectively to find anomalies.

## Solution

Splunk forwarders enable users to collect the Windows infrastructure's Sysmon data from the endpoint in real time. Splunk software automatically transports the events that are relevant for analyzing anomalies to the endpoint.

The Splunk platform provides two key functions to solve the challenges of making the best use of sysinternal events for detecting early signs of known advanced malware infections:

1. Collections of Windows activities: The Splunk Windows OS-based forwarder to easily collects all sysinternal data through event logs
   - Provides a simple agent for collecting all Windows data (event log, sysinternal, perf mon, files)
   - Allows secure and highly confident transport means for centralizing data in an analytics platform
   - Sysmon specific formatting and process ability to immediately apply analysis
2. Analytics base for searching and analyzing anomalies: Using simple search, statistical summation and calculation to highlight rare values in process creation details.
   - Pivots into different endpoint criteria to dynamically derive results
   - Applies machine learning

By applying an analytical approach to the data, the Splunk platform allows users to identify abnormalities in the activity endpoints by eliminating a normal pattern in statistical calculation. The use of this technique can be widely used with 1) any Windows based server infrastructure 2) or by collecting sysinternals from all Windows clients. This use case can be applied to the majority of security operations. Regardless of whether the organization already has an endpoint security solution or not, the wealth of information provide significant value to assess the security of an endpoint. There also could be other uses of the sysinternal where it will add more context to either IT operations and service analysis.

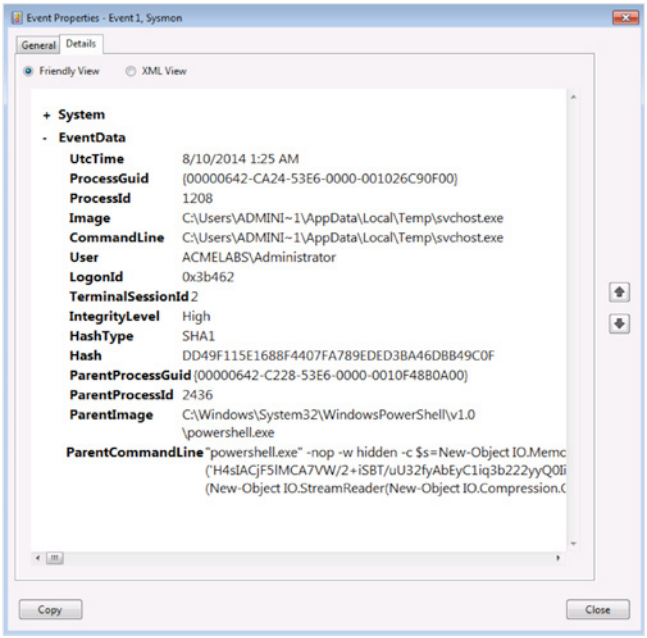
## Data Sources

Data sources that are required to detect the potential activities of malware on Windows Endpoint is sysinternal collected through Windows event log using Sysmon. An organization can gain detailed information by installing Sysmon provided by Microsoft, then installing Splunk forwarder to define what needs to be collected and filtered. This sysinternal data is where finding the indications of odd activities would begin, but additional correlation to trace the how and what got infected; further ingesting proxy, IDS/IPS, DNS/ stream data is recommended to root case the route of a potential infection and determine the scope and mitigate the incident. Analyzing the sysinternals through Splunk software would provide definitive indications of compromise in detecting potential of any malware, whether it's known or unknown.

- Windows Sysinternals using Sysmon through event log (required)
- Proxy, IDS/IPS, DNS, stream (recommended for further investigation beyond detection)

Event log with Sysmon installed provides the following details to be collected in Splunk software:

- Process creation including full command line with paths for both current and parent processes
- Hash of the process image using either MD5, SHA1 or SHA256
- Process GUID that provides static IDs for better correlations as opposed to PIDs that are reused by OS
- Network connection records from the host to another includes source process, IP address, port number, hostnames and port names for TCP/UDP
- File creation time changes
- Boot process events that may include kernel-mode malware



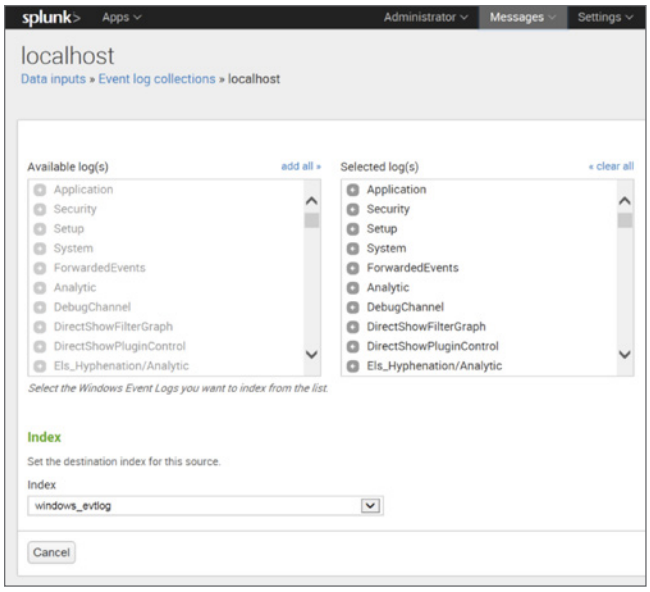

Example of Windows event log through sysmon

## Collection of Windows Activities Events

Collecting various pieces of information from the Windows infrastructure is easy with the Splunk forwarder.

Here are a few simple steps to collect and integrate Sysmon data into the Splunk platform:

1. Install Sysmon on your Windows-based endpoint, which can be downloaded from the following link: **https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon**
2. Install Splunk forwarder on the endpoint and it will forward sysinternal messages in real time to a Splunk instance
3. Install Splunk Add-ons for Microsoft Sysmon and easily configure Splunk to extract and map to CIM. Download it here: **https://splunkbase.splunk.com/app/1914/**




Once Sysmon is installed, you can use Splunk's "data inputs" to decide what you want, just select the type of event logs to transport to the Splunk Indexer.

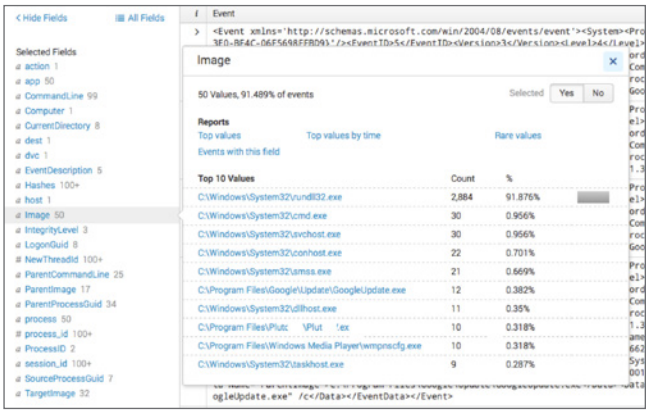
Now that you have events in the Splunk platform there is a wealth of information available to you. The basic search to call the sysinternal events from Splunk index is:

```
sourcetype="XmlWinEventLog:Microsoft-Windows-Sysmon/Operational"
```

The following is an example of data collected in Splunk software. Windows event log format is converted into XML combining all different fields into a single line event.

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Sysmon' Guid='{5770385F-C22A-43E0-BF4C-06F5698FFBD9}'/><EventID>1</EventID><Version>5</Version><Level>4</Level><Task>1</Task><Opcode>0</Opcode><Keywords>0x8000000000000000</Keywords><TimeCreated SystemTime='2016-02-04T01:58:00.125000000Z'/><EventRecordID>73675</EventRecordID><Correlation/><Execution ProcessID='1664' ThreadID='185 6'/><Channel>Microsoft-Windows-Sysmon/ Operational</Channel><Computer>FSAMUELS</Computer><Security UserID='S-1-5-18'/></System><EventData><Data
```

Data collected in XML format with sysinternal events are all parsed into fields in the Splunk platform with the help of the Splunk Add-on for Sysmon. Browsing through complex sysinternal events is now easy, just point and click on parsed fields.




## Searching for Process Creation Anomalies

The challenge is, how do we protect against the unknown? Unknown here means that there is no list to verify against things that are not just defined either right or wrong, but what's right or wrong derives from the data itself. It is based on calculated results with the understanding of what is the majority versus the minority and associates other analytical details related with them.

## Objective of the Analytics Approach

Detecting changes of activities entails finding anomalies by comparing what happened in the past to what is happening now.

The elements to validate different aspects of determining anomalies are:

- What is pre-existing and new?
- What are the statistics on pre-existed versus new to validate which is old (being normal) and new (as something that needs to be validated)?
- What are the time relations of existed and new entities?
- The association between an existing entity and other entities, such as the number of assets associated with it.

Utilizing insights related to validating anomalies, now we can eliminate the normal to filter out the anomalies that are most likely to be evaluated and analyzed.

These kinds of distinctions are possible when the statistics of different entities are compared to each other.

Windows Sysinternal provides extensive detail into understanding the status of endpoints in terms of endpoint security and vulnerability. One of the notable powers of analyzing sysinternals is the ability to gain visibility into what processes and files are installed and executed. There are events related to the execution of processes, indicating activities on the system which provides critical sources of information to help security analysts understand:

- What processes have been executed
- What is the directory origin of the executable
- What is the parent process that executed the executable
- What is the fingerprint of the executed process

All of these insights gained from the sysinternals are a critical part of collected system activity information in applying analytics to find anomalies of processes and action executed in an endpoint. This is an easy task with the data collected from the different Sysmon sources. Using Sysmon's hash information attached to each process creates events as MD5, SHA1 or SHA256, and an analyst can identify different versions of a certain system executable.

For example, why do we care about the full path of a process "cmd.exe?" Even though the "cmd.exe" is a legitimate looking executable on Windows, we can see the odd path of the binaries, potentially linking it to a "black sheep." How about the MD5 hash of the binary "cmd.exe" that is different from all the other "cmd.exe" in the network? This is a clear indication of file manipulation, potentially malicious code hiding as a legitimate executable.

## Malware Process Hiding as Existing OS or Application Process

Most PC users have experience looking at Windows process monitor, finding no particular problems where the OS seems to be running all the normal processes. Regardless of who may appear to be the user, we know that the PC is infected with all kinds of malware. An example of a "black sheep" malware disguising itself as a normal OS process is when malware processes run as if they are normal processes. How could this kind of "black sheep" be detected?

What about in the case of advanced malware, for example, a type malware that has never been known or detected by an anti-malware software product? This type of malware would be executed on an endpoint limiting the ability of most anti-malware detection software to raise a red-flag because the signature of the new executable is not known. Could this kind of problem be tackled using analytics? Analytics that compare a set of criteria from different executable fingerprints.

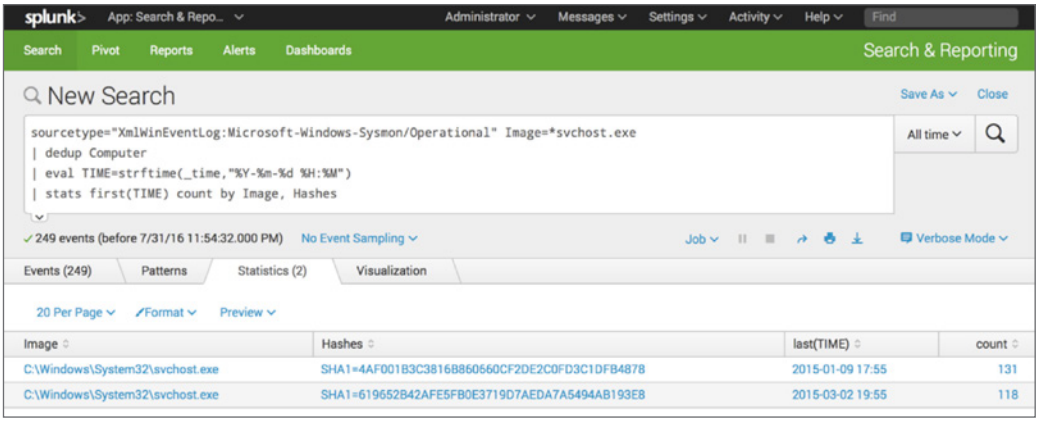
In order to find this, hashes on the Sysmon event play a key role. The hash information that gets attached to the Sysmon process creation event represents a unique fingerprint of an executable. If we were to find out what those existing fingerprints of trusted executables were versus comparing the new fingerprint for a similar executable that started recently, we can find the processes that are anomalies. This detailed Sysmon event about created processes and their associated hash can be analyzed with simple Splunk SPL summation by executable name.

This lists unique counts of executables regardless of how the executables are disguising themselves. A fingerprint of a hash means a non-arguable unique file or executable executed. On top of that sum the count of those unique hashes indicates what needs to be looked at more closely.

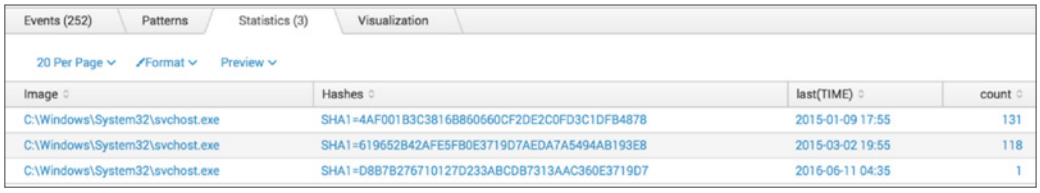
Search Syntax Below:

```
sourcetype="XmlWinEventLog:Microsoft-Windows-Sysmon/Operational" Image=*svchost.exe
| dedup Computer
| eval TIME=strftime(_time,"%Y-%m-%d %H:%M")
| stats first(TIME) count by Image, Hashes
```

The search to find all the same executable names with different hashes.

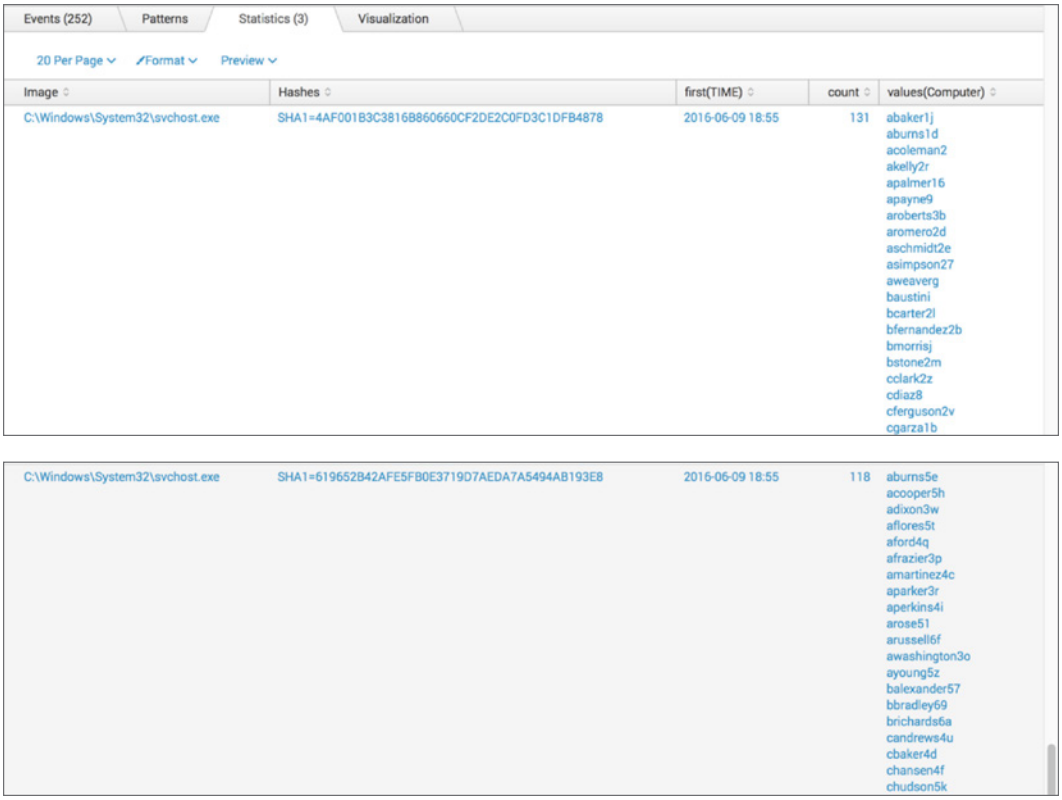



Based on the result of the search, the same executables svchost.exe with the exact same paths were found, but notice the hashes are different. This means that there are two variants of Windows OS, because this infrastructure is running a good balance of hosts that are ~~Windows 7 and Windows 8.~~ This seems normal because given the size of the network with more than 200 hosts, the distribution of hashes for a critical system process "svchosts.exe" is distributed at the quantity of each Windows version. Notice the sum of the instances, knowing the basic facts about the infrastructure running two versions of OS and seeing a good count of both results, we can conclude that things look normal.




In the following example, imagine the same search returns as in the previous example. The result shows a similar number of distributions for the first two majority hash executables, but it shows the third one with fewer hosts with a new SHA1 hash found. This means that the same executable with a different hash and a significantly lower number of process creations means this is a new executable executed with the same name as a system binary. The sum of counts of "1" indicates it's a rare frequency, not likely to be seen as a system executable unless we have another new version of OS with different system executables running on the network. If this is not the case, then this is a suspicious hash that needs to be referenced against a Google search.

Additionally, the "first(TIME)" function indicates the first time the anomalous executable was created and indicates that it is definitely a new process compared to the normal svchost.exe executables created before. The first time function provides insight into what existed versus what is new and correlating the sum of counts determines what is abnormal. The third hash and newer timestamp executable with a minor number of occurrences is most likely malware that potentially an anti-virus program didn't detect.




Make sure to verify what hosts are associated with the hashes for two different normal svchost.exe, as well as which hosts are involved in potential malware activities. This can be accomplished by listing unique values in the "computer" field from Sysmon data, using the values (Computer) function.

```
sourcetype="XmlWinEventLog:Micro
soft-Windows-Sysmon/Operational"
Image=*svchost.exe
| dedup Computer
| eval TIME=strftime(_time,"%Y-%m-%d
%H:%M")
| stats first(TIME), count,
values(Computer) by Image, Hashes
```

After analyzing a process with new hashes, we can conclude a couple of conditions to define a potential malware sneaking in as a system process:

- The process may look normal from the path and name of the executable, but the hash of the new executable in comparison with existing historical hashes are different.
- The frequency of process creation in contrast with an existing executable hash is significantly different.

Understanding the nature of the manipulation tactics, we can define a query that filters automatically by applying calculated steps that consider the quantitative contrast of process creation count with existing and newly executable hashes. You can use "eventstats" to calculate the sum of all occurrences and apply this to calculate the percentage of occurrences of each individual executable to make it easy to define a relative threshold that would pick out the "odd executables," even those that are masking themselves as sheep.

Search Syntax Below:

```
sourcetype="XmlWinEventLog:Micro
soft-Windows-Sysmon/Operational"
Image=*svchost.exe
| dedup Computer
| eval TIME=strftime(_time,"%Y-%m-%d
%H:%M")
| stats first(TIME) count by Image, Hashes
| eventstats sum(count) as total_host
| eval majority_percent=round((count/
total_host)*100,2)
```

Now, how do we define a search (rule) to have Splunk software look for these kinds of odd executables?

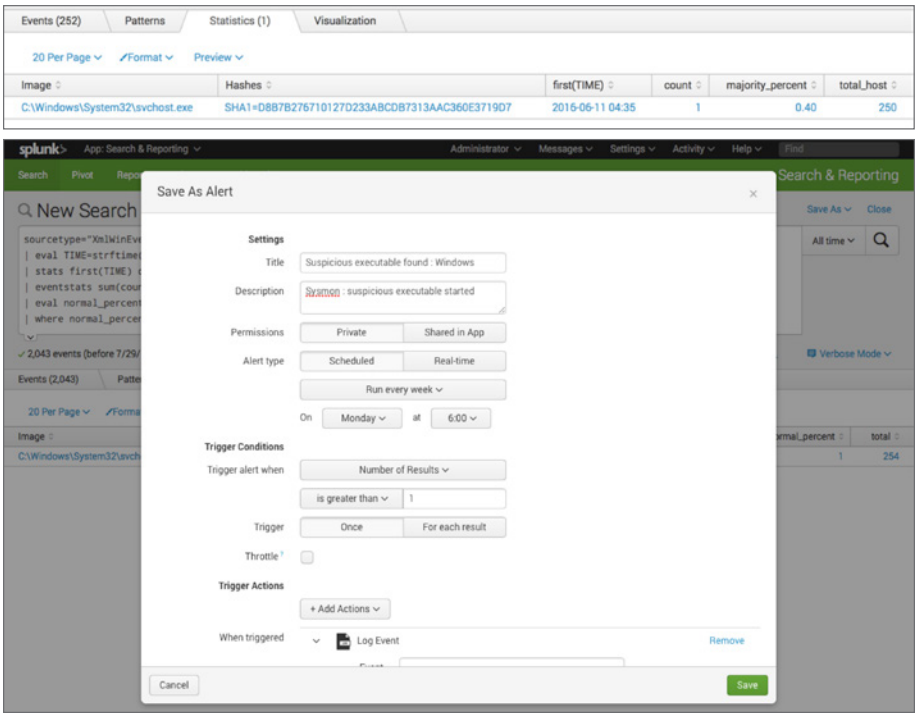
Expanding upon the previous relative quantity calculation and applying the "majority_percent<5" will eliminate the normal groups and expose the anomalous executable group based on relative threshold.

```
sourcetype="XmlWinEventLog:Microsoft-Windows-Sysmon/Operational"
Image=*svchost.exe
| dedup Computer
| eval TIME=strftime(_time,"%Y-%m-%d %H:%M")
| stats first(TIME) count by Image, Hashes
| eventstats sum(count) as total_host
| eval majority_percent=round((count/total_host)*100,2)
| where majority_percent<5
```

This kind of recipe can be applied to Splunk Enterprise's saved search or Enterprise Security's correlations search feature to do the analysis for us and automatically send the analyst alerts on the anomalous processes that could start up in any one of the Windows workstations running on the network.

## Summary

By using Splunk Enterprise and Microsoft Sysmon, security analysts can gain a significant understanding of detailed activities on endpoint, as well as the ability to detect advanced and unknown malware activities. Statistical analysis of detailed endpoint data risk in quantitative values for analysts to easily profile behavior of compromised hosts by adversaries and further define rules based on those values as threshold. This empowers security analysts to apply similar techniques to solve problems and use cases that could be addressed by only an analytical approach. Analytical approaches that contextually distinguish the differences and anomalies provide the security operations to detect advanced threats faster in order to ultimately minimize business impact.




Learn more about combating malware and ransomware by exploring security investigation use cases in Splunk's free, online demo environment.