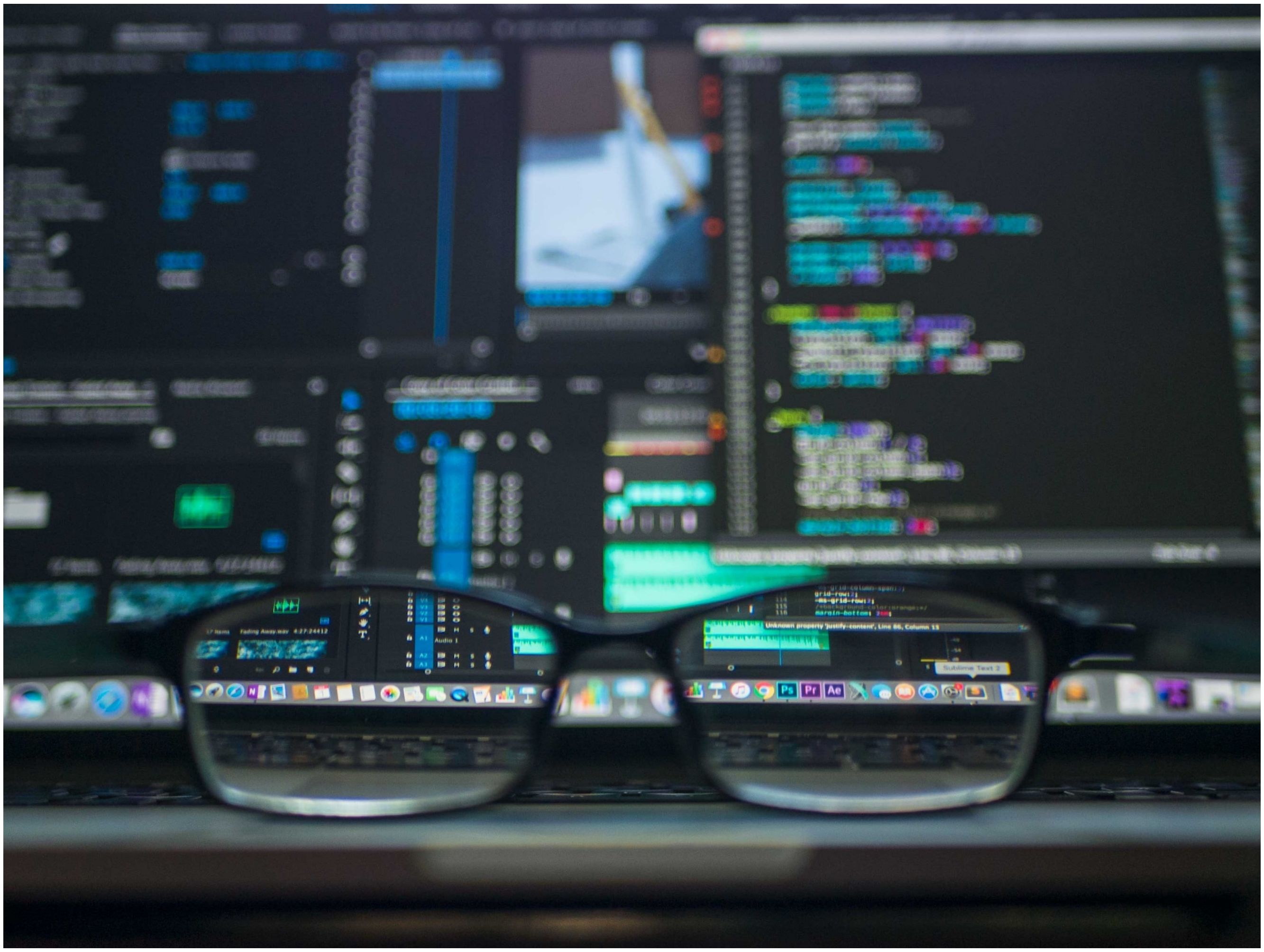
SUPERNOVA Redux, with a Generous Portion of Masquerading

splunk.com/en_us/blog/security/supernova-redux-with-a-generous-portion-of-masquerading.html

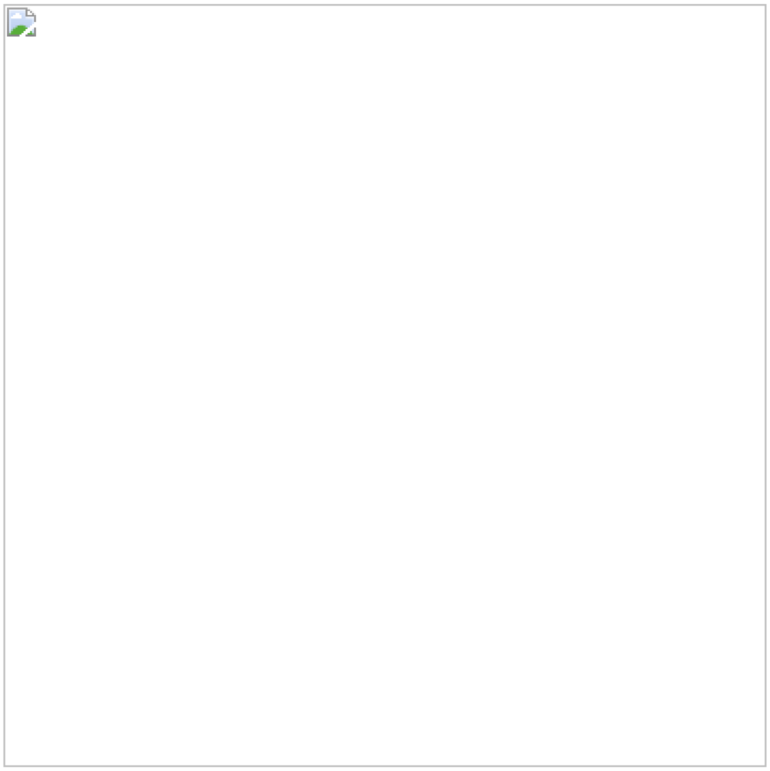
April 22, 2021



By John Stoner April 22, 2021

Contributors: Mick Baccio, Katie Brown, James Brodsky, Drew Church, Dave Herrald, Ryan Kovar, Marcus LaFerrera, Michael Natkin and John Stoner

If you want just to see how to find masquerading, skip down to the "*detections*" sections. Otherwise, read on for a quick breakdown of what happened, how to detect it, and MITRE ATT&CK mappings.



Recent SUPERNOVA Attack, Now with Masquerading

The Cybersecurity and Infrastructure Security Agency (CISA) issued an analyst report (AR21-112A) on April 22, 2021 that discussed a recent incident that they supported. As you read the first paragraph, it hits recent hot buttons: Pulse Secure and SolarWinds. Then you start wondering where is this going?

> “The threat actor connected to the entity’s network via a Pulse Secure virtual private network (VPN) appliance, moved laterally to its SolarWinds Orion server, installed malware referred to by security researchers as SUPERNOVA (a .NET webshell), and collected credentials.”
> — *Analysis Report (AR21-112A)*

All of a sudden, we see SUPERNOVA and we breathe a sigh of relief; we’ve got this, in fact, we blogged about this back in January 2021 in **Detecting Supernova Malware Solarwinds Continued**. But as we read farther, we come to find out that the adversary lovingly decided to take their copy of procdump.exe — a command line tool that is used to create dumps of processes and has been used by various actors to dump credentials — renamed it Splunklogger.exe, and placed it on the compromised SolarWinds server.

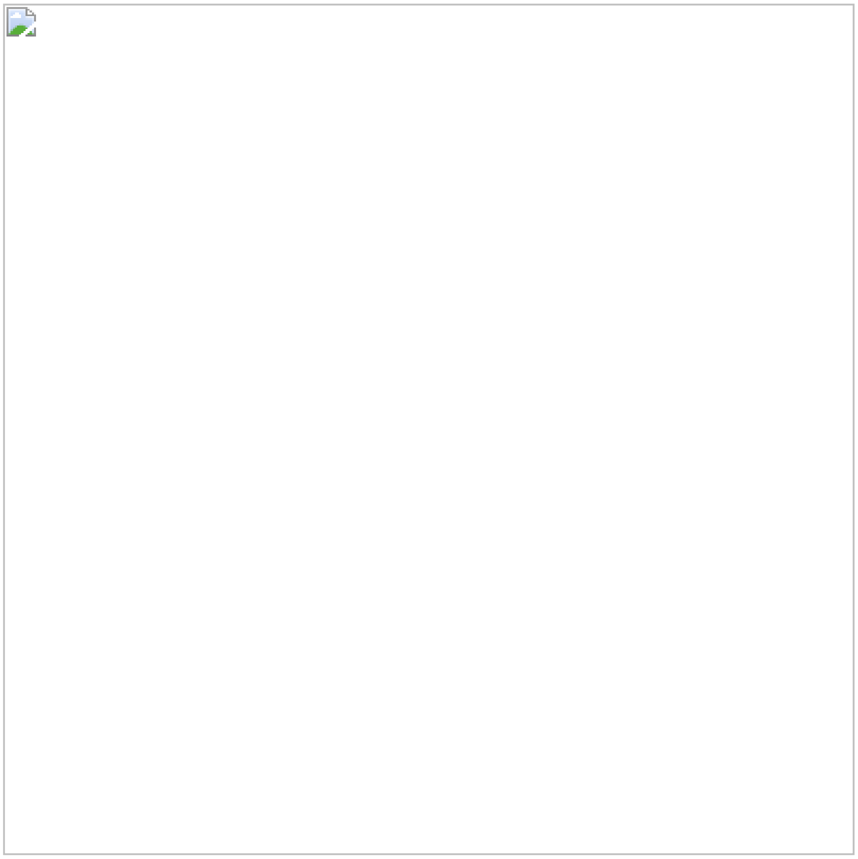
Great.

This blog will highlight some new detections that were seen in this attack along with a discussion around masquerading. We will also provide some detections that you can take advantage of in your own environment. As mentioned before, we won’t go deep into SUPERNOVA itself as we already have covered that in a previous blog, but these actions on objective are important to call out.

What You Need to Know

This specific attack has a few interesting traits. The first is that the adversary is using residential IP addresses based in the United States (US) to make them appear as US-based employees and then leveraging valid accounts to gain access via the VPN.

From there, the adversary used a virtual machine and obfuscated PowerShell scripts to move laterally to the SolarWinds server. At this point, the SUPERNOVA webshell is installed. Due to logs being cleared during the attack, CISA was not able to determine if the adversary exploited CVE-2020-10148, an authentication bypass vulnerability of SolarWinds Orion or another method to gain access.



At this point, the adversary is collecting credentials as well as deploying tools to maintain persistence, evade defenses and other activities. A common tool that certain adversaries use is procdump.exe. Procdump.exe is a Microsoft command line utility that is used to monitor applications and can create crash dumps. Adversaries have been observed using procdump to dump credentials. To obfuscate the existence of procdump.exe on the SolarWinds server, the adversary renamed their copy of procdump.exe to splunklogger.exe. This masquerading technique is fairly common with certain utilities because the existence of that utility on certain systems may trigger alarms for organizations, whereas a tool like Splunk is used in many organizations and would raise less concern when seen.

After credentials were dumped from LSASS memory, the adversary used the organization’s web server to exfiltrate the credentials and then deleted the web server logs in an effort to cover their tracks.

Additional access occurred after the initial attack by leveraging credentials that were likely cracked offline from the initial credential dump. A final identified access event occurred where both procdump.exe and winrar.exe were seen masquerading as wininit.exe and the adversary made an attempt to archive credentials, probably before they were exfiltrated.

Detecting Masquerading As Well as Indicators of the SUPERNOVA Attack in Splunk

Here we will give you some hot-off-the-press searches to help find some of the badness derived from the CISA Analysis Report on this recent SUPERNOVA attack. If we have coverage for these searches in Splunk security content, we call them out further below in the MITRE ATT&CK section.

We covered some thoughts on detecting the SUPERNOVA webshell in our previous post on the subject as well as the associated vulnerabilities, so today we will focus on the activities that took place after the webshell was established, specifically around masquerading and file integrity of the files manipulated.

Indicators of Compromise (IOCs)

CISA published IOCs, including file names, hashes and IPs, in their blog post. So we collected the common hashes for procdump.exe, along with the IOCs that CISA identified and converted these indicators into simple CSV format so that you may use them as lookup tables — they are posted here. But what’s a lookup table, and how does it help with security detection in Splunk? Got you covered there, too.

Process Monitoring

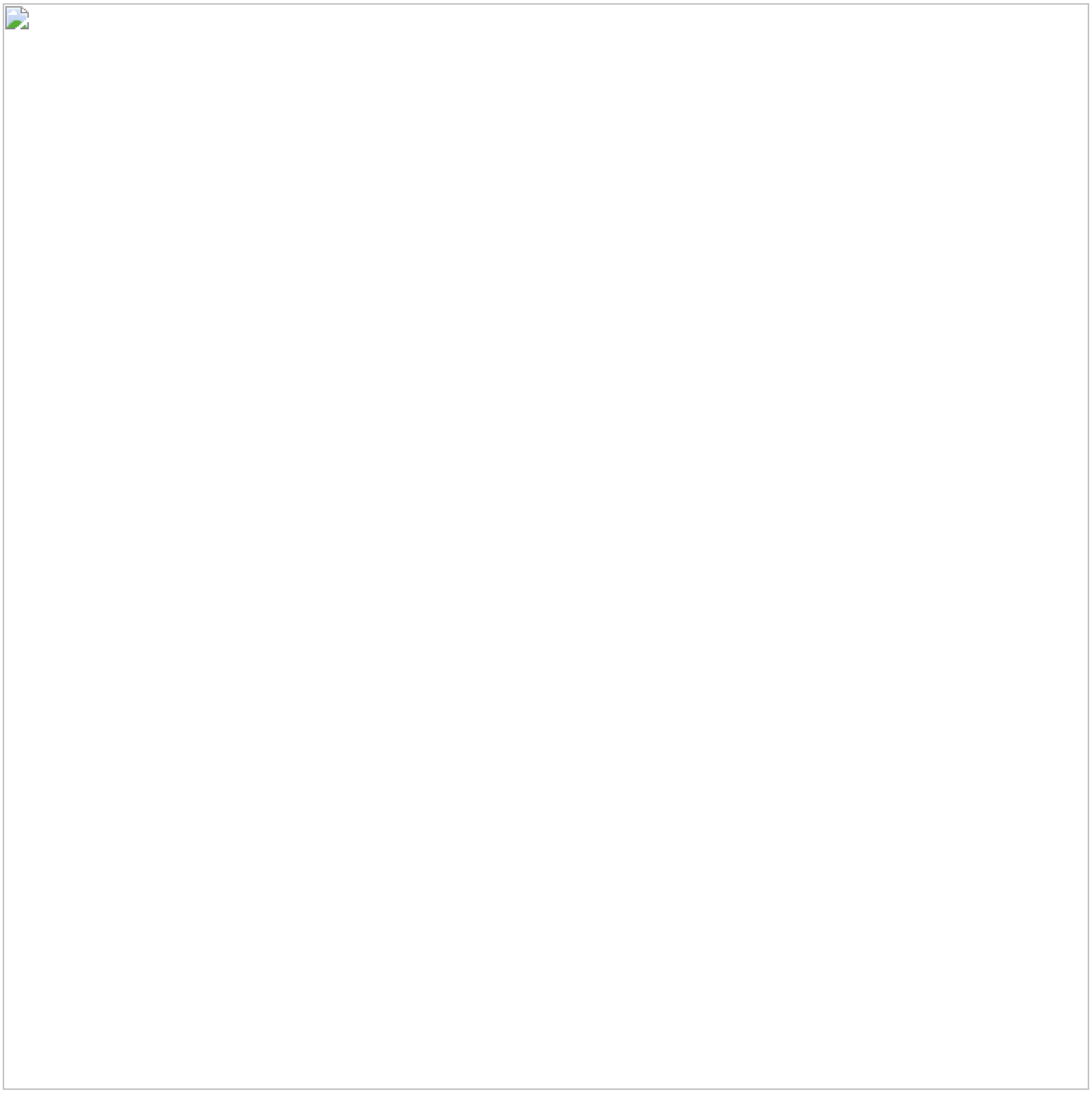
We frequently are asked "why should we use Sysmon for process monitoring instead of native Windows capability via Event ID 4688?" This situation is a perfect example as to why the rich data gathered by Sysmon is so valuable to defenders and threat hunters: hashes.

Microsoft's documentation for 4688 is available here. If we look in the example Event XML, we will only see information such as the NewProcessId or NewProcessName. If we've turned on CommandLine tracking, we'll be provided that information as well.

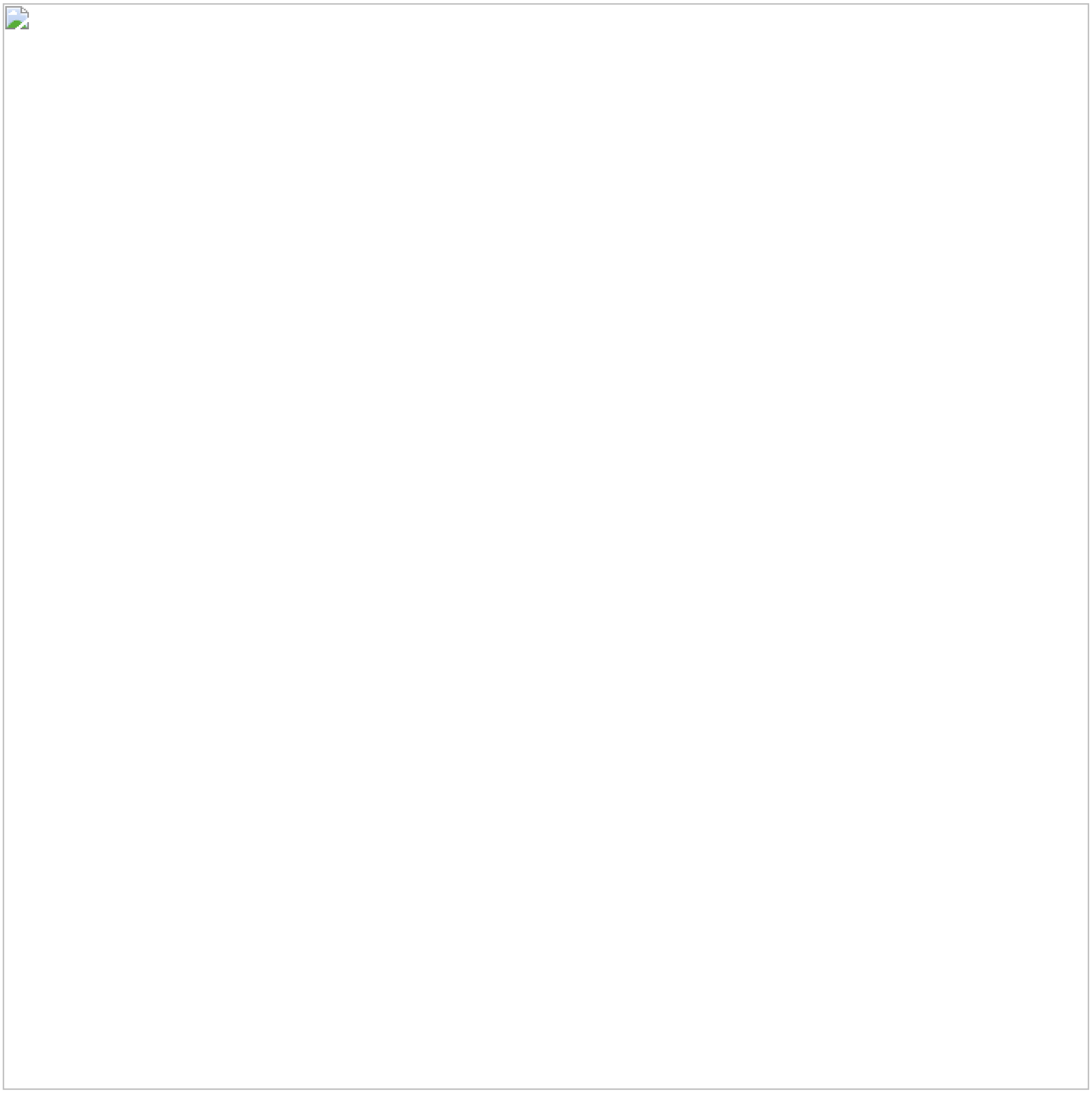
Comparatively, Sysmon Event Code 1 has a number of other fields including multiple types of hashes, the Company, and even the parent process id to better contextualize the process that was created.

If we compare files being executed based on the name of the original file and the process, we can use Sysmon data with a search like this to get a side by side comparison and use the match function with the eval command to get a comparative.

```
EventCode=1 OriginalFileName=* process_name=*

| eval OriginalFileName=upper(OriginalFileName) | eval process_name=upper(process_name)

| eval match=if(OriginalFileName=process_name,"Match","No Match")

| search match="No Match"

| table _time host OriginalFileName process_name match
```



As you can see above, we can see that psexec.c is running under the name of smb.exe. It is important to note that without additional filtering, the search above is a bit noisy. However, for our purposes, we could easily swap out the OriginalFileName value of * in this search to look for just procdump.exe or add the string “*splunklogger*” like the search below.



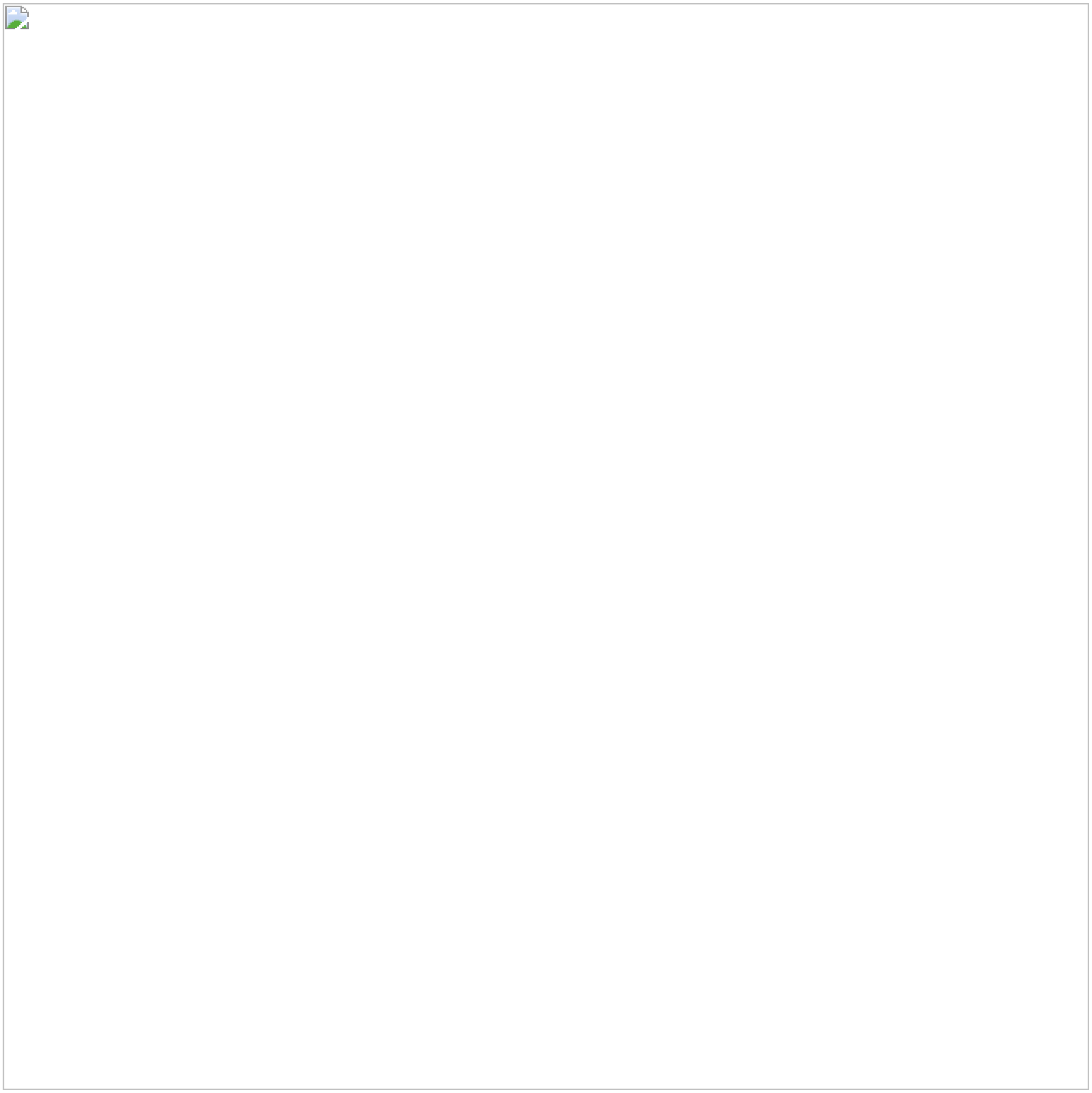
Another technique that we can use to identify masquerading is the use of file hashes. To do this, we can again use a favored capability of Splunk, that is the lookup!

We mentioned them above, and just to make sure this blog is super long, let’s cover them in greater detail. A very effective way of determining whether or not a process executing on one of your production servers is legit is to use lookups sourced with “known good” metadata about processes normally found in your environment. Assuming you have tight control over upgrades that will change legitimate binaries (e.g. a change control process/board), any executing binaries on your production servers that are “unknown” when compared to “known good” can be flagged.

Here’s one way of accomplishing that in Splunk. First, we leverage Sysmon’s Event Code 1, which provides hash values and a lot of other interesting process metadata, to harvest this data from a known-good “golden image” server in the environment, and we pipe it to a lookup called UFKnownLookup.csv. Note, there’s all sorts of interesting metadata in these events, like the company name, the version number of the file, a description, and so forth:

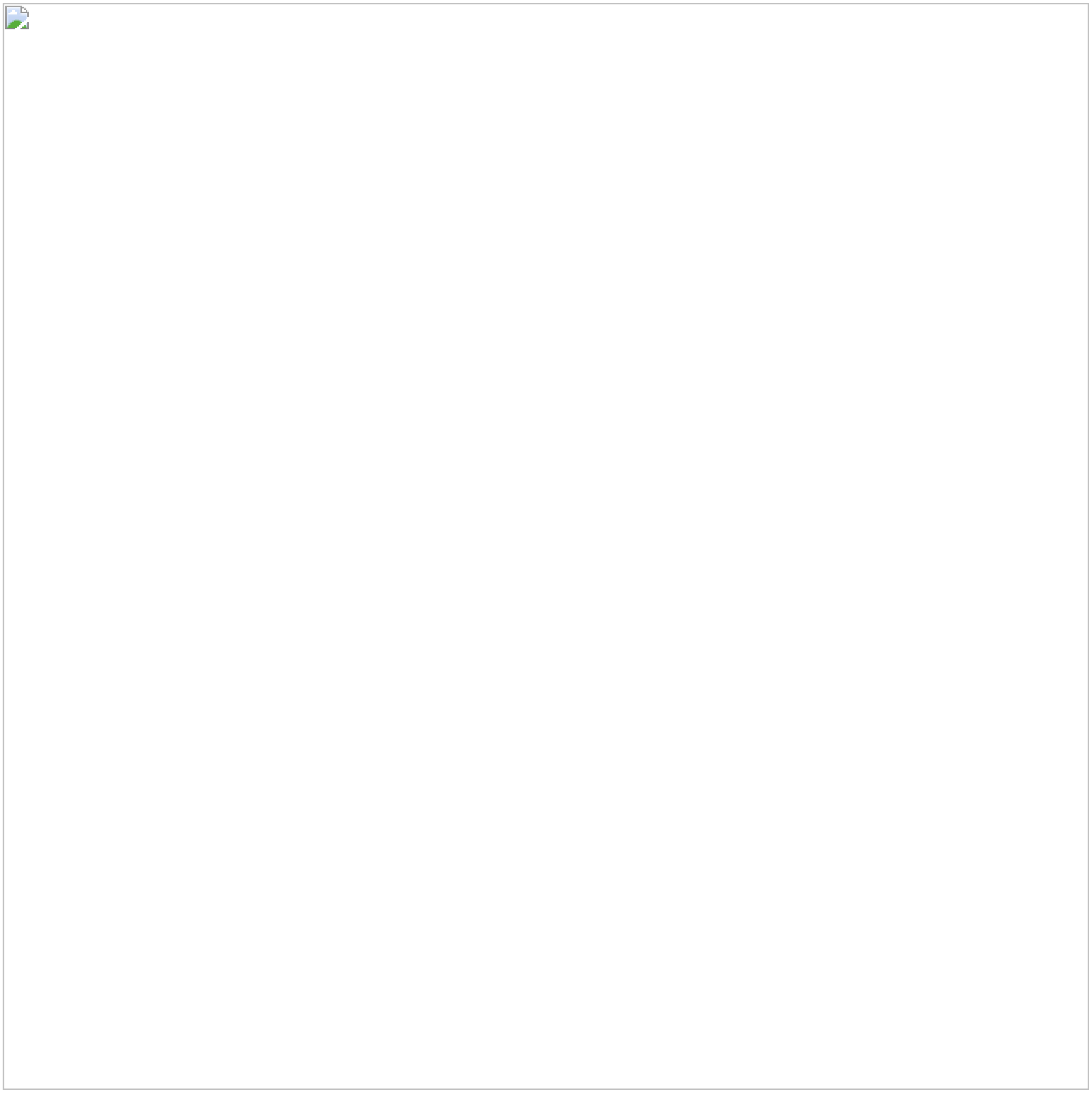
```
index=endpoint process_name=splunk* EventCode=1 host=frostbite
| eval knowngood=1
| stats values(process_name) as process_name values(Company) as vendor values(Description) as
description values(FileVersion) as version values(knowngood) as known_good by SHA256,MD5
| outputlookup UFKnownGood.csv
```

Now, we can leverage it in subsequent searches. For example, let’s rip a page from the CISA SUPERNOVA report, take a copy of procdump.exe (which is a Sysinternals binary) and rename it “splunklogger.exe” and then put it on our desktop and run it. It will look like this in a Sysmon event.



Now, let’s run a search to find Splunk binaries that are “unknown” to us, leveraging the lookup:

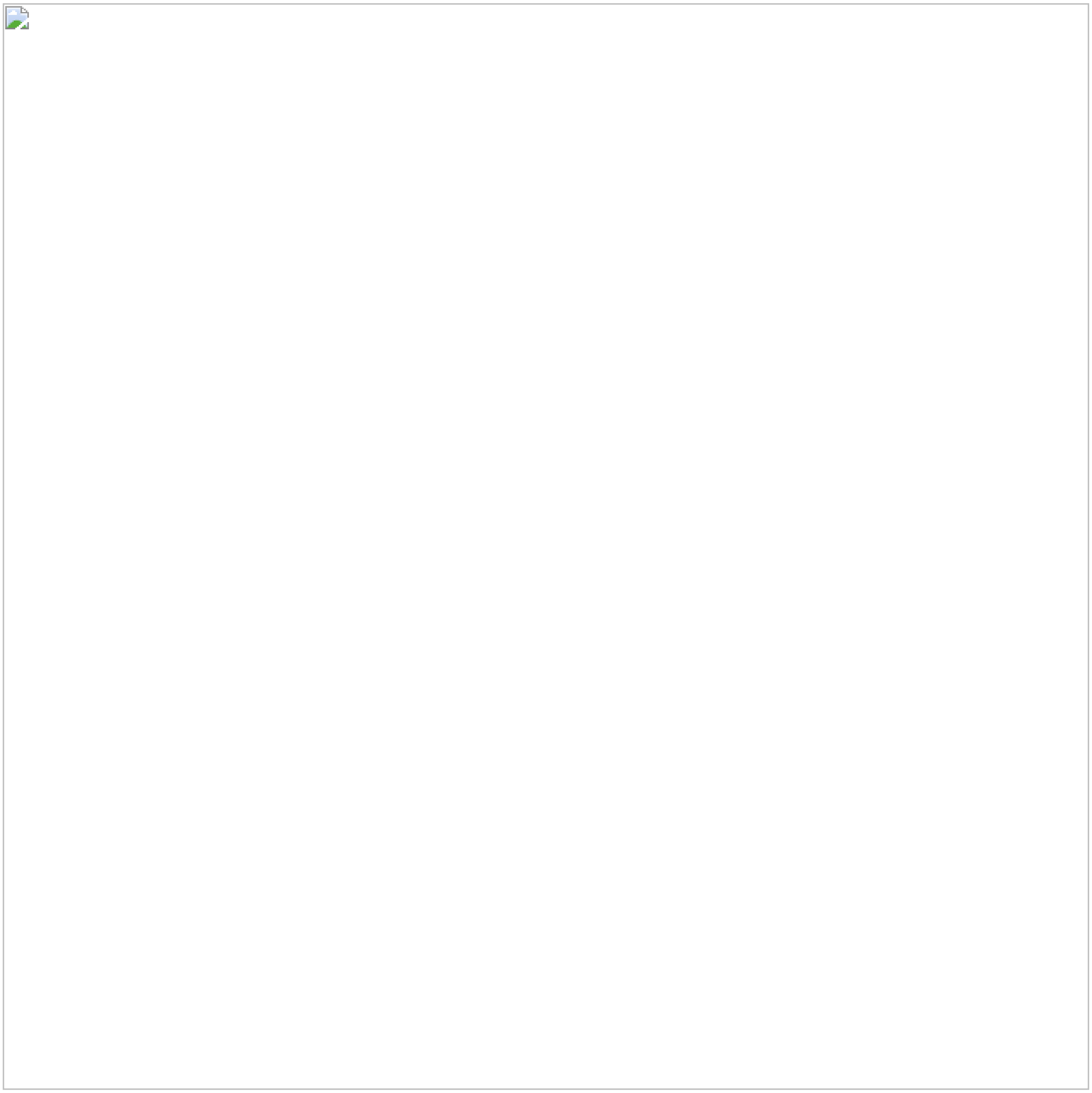
```
index=endpoint process_name=splunk* EventCode=1 host=frostbite

| lookup UFKnownGood.csv SHA256 OUTPUT known_good

| eval known_good = case(known_good == 1, "1", 1=1, "0")

| search known_good=0
| stats values(process_name) as process_name values(Company) as vendor values(Description) as
description values(FileVersion) as version values(known_good) as known_good by SHA256,MD5
```



And **voila**! We can see that a binary called “splunklogger.exe” executed, but it wasn’t in our approved lookup list, and oh, by the way, even though it is called “splunklogger.exe” it certainly isn’t a real Splunk binary, based on the Company and Description metadata.

In production, generating the lookup itself against raw data is a reasonable thing to do on an occasional basis. But for matching voluminous Sysmon data in Splunk, a tstats search against an accelerated data model from the Common Information Model is more optimal, like this one against Endpoint.Processes:

```
| tstats summariesonly=t prestats=t count,values(Processes.process),values(host) as host from
datamodel=Endpoint.Processes where (Processes.process="*splunk*.exe*" AND
Processes.process!="*cmd.exe*") by Processes.process_hash

| rename Processes.process_hash as hashes

| stats values(*) by hashes

| rex field=hashes "SHA256=(?<SHA256>.*),"

| lookup UFKnownGood.csv SHA256

| eval known_good = case(known_good == 1, "1", 1=1, "0")

| search known_good=0
```



Fee Fi Fo FIM

We also figured file integrity monitoring (FIM) was a super appropriate topic to cover since, ya know, procdump.exe went all splunklogger.exe on us. The goal we're trying to accomplish with FIM is simple: detect unauthorized changes made to files, directories, network devices, OS and more. This can be accomplished by establishing a "baseline" for a file state, and monitoring for changes made to that state. It's a great way to quickly identify file discrepancies, modifications, and additions.

Need a FIM solution now? There are multiple software solutions that are designated for file integrity monitoring like Tripwire and Qualys FIM.

More of an open source kind of person? Not a problem, there are several solutions available depending on your requirements. Some of the more popular open source FIM solutions include OSSEC and osquery.

Got none of that? You can always use “native” file activity monitoring, from things like Sysmon’s FileCreate and FileDelete events, or even Windows 4663 events. All of the solutions we mention integrate well with Splunk, and many populate our Change data model for ease of use.

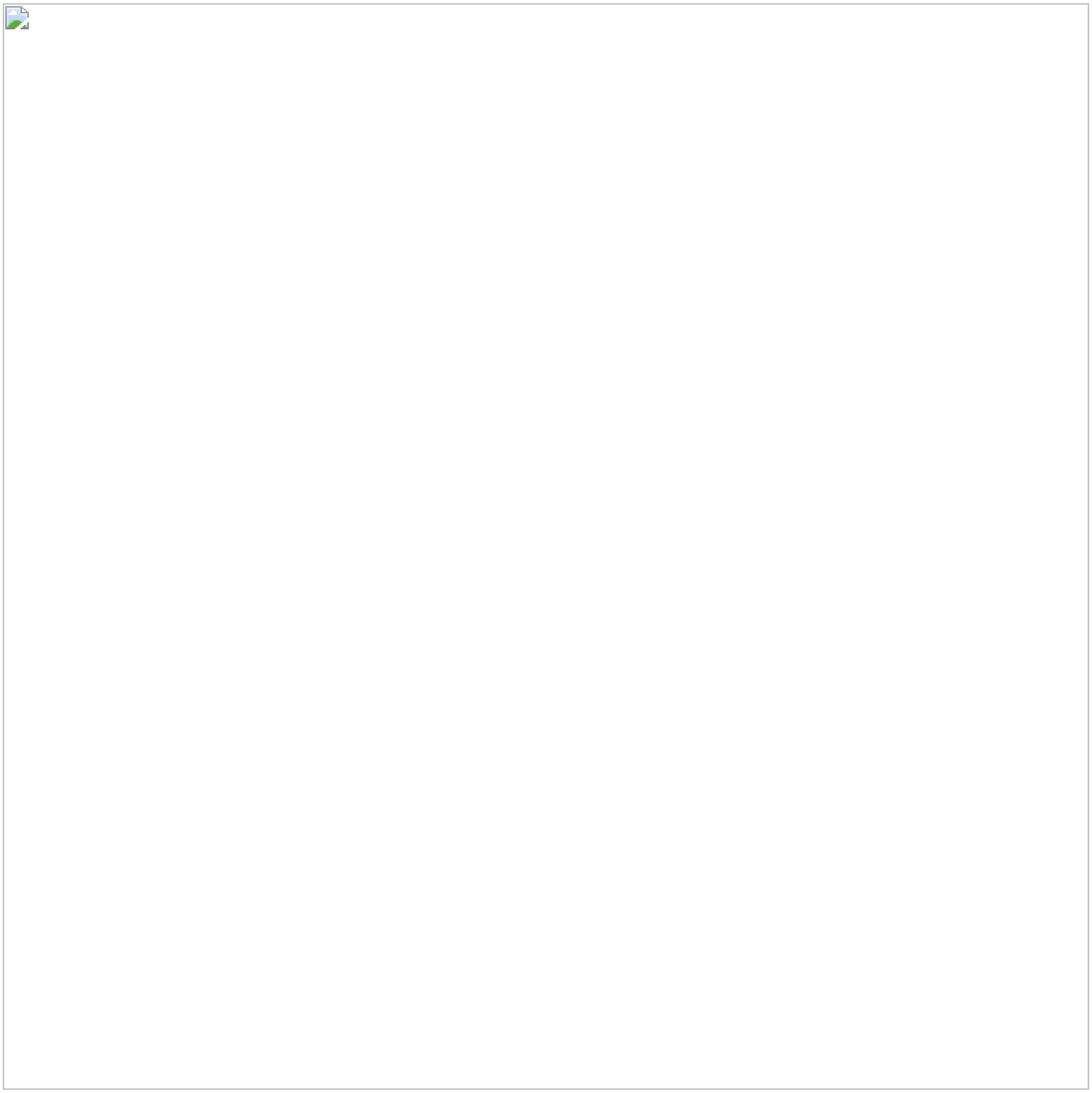
Splunk Enterprise Security and ESCU

Threat Intelligence Framework

If you are using Splunk Enterprise Security, the lookups of IOCs that are listed above can be ingested easily into the threat intelligence framework. Perhaps you aren’t sure how to do that. No worries, we published some guidance and a how-to on integrating lists of IOC into the Enterprise Security threat intelligence framework.

Enterprise Security Content Updates (ESCU)

For folks using ESCU, our Threat Research team already has a number of detections around masquerading. While they are not all in a single analytic story, they can be found by using the Keyword Search. In fact, if you check out the MITRE ATT&CK table below, you can cut and paste those Splunk Search titles into the Keyword Search (place * between the words in place of spaces) to view them in ESCU. If you have ESCU running today, you already have some great coverage!



MITRE ATT&CK

Reviewing the CISA Analysis Report (AR-21-112A), we mapped the adversary’s activity to MITRE ATT&CK. Each tactic is then linked to Splunk content to help you hunt for that information. Be aware; these searches are provided as a way to accelerate your hunting. We recommend you configure them via the Splunk Security Essentials App. You may need to modify them to work in your environment! Many of these searches are optimized for use with the tstats command.

Finally, as more information becomes available, we will update these searches if more ATT&CK TTPs become known.

ATT&CK Technique	**Technique/ Sub-Technique Title**	**Splunk Searches**
T1105	Ingress Tool Transfer	Suspicious Curl Network Connection
T1036.003	Rename System Utilities	System Processes Run From Unexpected Locations
T1505.003	Web Shell	Detect Exchange Web Shell W3WP Spawning Shell Supernova Webshell
T1078	Valid Accounts	Reconnaissance of Access and Persistence Opportunities via PowerSploit modules Setting Credentials via DSInternals modules Probing Access with Stolen Credentials via PowerSploit modules Setting Credentials via PowerSploit modules Reconnaissance of Credential Stores and Services via Mimikatz modules Reconnaissance and Access to Accounts and Groups via Mimikatz modules Reconnaissance of Privilege Escalation Opportunities via PowerSploit modules Applying Stolen Credentials via Mimikatz modules Applying Stolen Credentials via PowerSploit modules Setting Credentials via Mimikatz modules
T1047	Windows Management Instrumentation	Script Execution via WMI Process Execution via WMI Remote Process Instantiation via WMI Reconnaissance and Access to Operating System Elements via PowerSploit modules WMI Permanent Event Subscription WMI Temporary Event Subscription
T1018	Remote System Discovery	Windows AdFind Exe

T1070.001	Clear Windows Event Logs	Windows Event Log Cleared Suspicious wevtutil Usage
T1021.002	SMB/Windows Admin Shares	Reconnaissance of Connectivity via PowerSploit modules Reconnaissance and Access to Shared Resources via PowerSploit modules Reconnaissance and Access to Shared Resources via Mimikatz modules Detect PsExec With accepteula Flag SMB Traffic Spike SMB Traffic Spike - MLTK
T1057	Process Discovery	Reconnaissance and Access to Processes and Services via Mimikatz modules Reconnaissance and Access to Operating System Elements via PowerSploit modules
T1083	File and Directory Discovery	Reconnaissance and Access to Operating System Elements via PowerSploit modules
T1140	Deobfuscate/Decode Files or Information	CertUtil With Decode Argument
T1003.001	LSASS Memory	Detect Mimikatz Using Loaded Images Dump LSASS via comsvcs DLL Create Remote Thread into LSASS Access LSASS Memory for Dump Creation Detect Credential Dumping through LSASS access Dump LSASS via procdump Creation of lsass Dump with Taskmgr Dump LSASS via procdump Rename
T1041	Exfiltration Over C2 Channel	Detect SNICat SNI Exfiltration

T1059.001	PowerShell	Malicious PowerShell Process - Connect To Internet With Hidden Window Set Default PowerShell Execution Policy To Unrestricted or Bypass Any Powershell DownloadString Malicious PowerShell Process With Obfuscation Techniques Any Powershell DownloadFile Malicious PowerShell Process - Execution Policy Bypass
T1105	Ingress Tool Transfer	Suspicious Curl Network Connection

Here is a list of all the MITRE ATT&CK TTP’s that we saw being used in this attack:

```
T1133, T1078, T1059.001, T1140, T1105, T1505.003, T1552.004, T1036.003, T1003.001, T1074.001,
T1041, T1070.001, T1021.002, T1047, T1057, T1036.005, T1560.001, T1083, T1018
```

Conclusion

This blog is a little bit of an outlier. We realize that attacks are continually occurring, but with the masquerading of procdump.exe as splunklogger.exe as well as the use of the SUPERNOVA malware, which was so recent, it felt like a good time to talk about this specific attack. Masquerading and obfuscation are capabilities that many adversaries use during their attacks. Hopefully, these searches will provide you the ability to have more visibility into your environment and any malicious activity that you might be experiencing. If they don’t work perfectly, think of them as “SplunkSpiration” :-). Again, you may have to modify them to work in your unique environment. If we uncover additional information, we will update this blog.

Posted by

John Stoner

-
-

I grew up in Virginia, graduated from Penn State and came back to Virginia where I have worked with databases and cyber for over 20 years. A job or two felt like I really was a cast member of The Office or Office Space, but every one of them taught me something new.