SPLUNKING THE ENDPOINT V:

**Enough Already! (SEC2007)**

brodsky@splunk.com
Director Global Security Strategists | Security Kittens

October, 2019
V1.0
whoami – @james_brodsky

- Director, Global Security Strategists (Louisville, CO)
- Lead a team of Splunk security strategists across the US, UK, Australia
- Have been involved with security here since my start
- .conf Splunking the Endpoint! for FIVE years
- BOTS 1.0, 2.0, 3.0, 4.0. BOTN 1.0, 2.0.
- CSC 20 Whitepaper, FFIEC Whitepaper (co-author), other compliance, Tripwire apps, blogs, Sysmon contributions, etc, etc....
It’s a hands-on session. Eventually. But first slides. Lots of pink slides.
Nope. Still can’t get Splunk to run on an Apple IIe. You need to be using a functional, modern computing device.

And it needs to be on the Internet.

And it needs a relatively modern browser.
Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

Splunk, Splunk>, Listen to Your Data, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2019 Splunk Inc. All rights reserved.
It is fun to assign folks to servers randomly…
Our BOTS adversary this year “Violent Memmes” is loosely based on APT 28/29 and Turla.
My thought process.
(not me.)
Hi James, your order is being shipped!

Your order will ship to: [Address]

Estimated delivery:
Mon, Sep 30 - Mon, Oct 7

1 Ruble = 100 kopeks

USSR SOVIET RUSSIAN 100 KOPEK COINS 1961-1991 COLD WAR HAMMER AN...
Total: $32.40
Item ID: 142829786034
Seller: abru.uk2013(3.740)
1 Ruble = 100 kopeks

$32.40/300 = 11 cents per kopek coin from eBay, September 2019
1 Ruble = 100 kopeks

$32.40/300 = 11 cents per kopek coin from eBay, September 2019

Historical value of former Soviet ruble in 1992 = $1.80 USD (or 18 cents/kopek)
1 Ruble = 100 kopeks

$32.40/300 = 11 cents per kopek coin from eBay, September 2019

Historical value of former Soviet ruble in 1992 = $1.80 USD (or 18 cents/kopek)

Cumulative Rate of Inflation from 1992-2019=82.9%, or 33 cents!
1 Ruble = 100 kopeks

$32.40/300 = 11 cents per kopek coin from eBay, September 2019

Historical value of former Soviet ruble in 1992 = $1.80 USD (or 18 cents/kopek)

Cumulative Rate of Inflation from 1992-2019=82.9%, or 33 cents!

300 kopeks should be worth $99!
...and I paid $32.40.
$32.40/300 = 11 cents per kopek

1 Ruble=100 kopeks

Historical value of former Soviet ruble in 1992=$1.80 USD (or 18 cents/kopek)

Cumulative Rate of Inflation from 1992-2019=82.9%, or 33 cents!

300 kopeks should be worth $99!

Are you kidding me?

Buy soviet kopeks on eBay

Exchange for US dollars

Make 200% profit…

I SHALL AMASS A FORTUNE
I could retire early!
Dear

For the past six and a half years I have enjoyed my time helping to bring value to security customers at Splunk. However, when perusing eBay over the weekend, I found a significant loophole involving the former Soviet kopek, and suddenly realized that I could very easily triple my money by purchasing massive quantities of the old coins and converting them into US Dollars.

Therefore, my last day with the company will be 10/18/2019. I’ll see what I can do in order to prepare someone to take over the Endpoint talk at .conf, and also someone else to write terrible questions involving arcane search commands in BOTS that everyone gets angry about.

Thanks for your support, and you can reach me in the future at kopek_kurrency@aol.com.

-jb
I bought a jet!
then I looked to see what happened to the ruble after 1992.
600 kopeks equals 1 cent.
300 kopeks that I bought = ½ a cent.
...but then....

I paid 3,240x more than their worth for 300 kopeks.

600 kopeks equals 1 cent.
300 kopeks = ½ a cent.
I returned the jet.

I groveled for my old job.
And here I am, presenting the FIFTH endpoint talk in as many years!

therefore...
We will **NOT** cover...

- What a Universal Forwarder is
- RAM Scraping POS Malware
  - Ransomware
  - Mac endpoints
- Why sysmon and osquery are awesome
  - Stranger Things
  - Endpoint forensics
  - Why everything is pink
- The difference between “pike” and “pipe”
  - John Denver
  - Machine Learning/AI
- Gluten free fortune cookies
  - Powershell Empire
  - Subverting Sysmon
  - Avocado Toast
  - Voltaire

All of these topics and more, in the .conf archives...

search “brodsky.”
But we **will** cover...

- What the latest endpoint surveys tell us & what Splunk has seen recently
- Alternatives to the UF, and Best Practices for commercial solutions
  - What NOT to do when you collect with the UF
  - New Stuff in Sysmon, Windows TA, etc…
- Endpoint Diet! Clever Event Reduction techniques
- An new way to guide which WinEvents to collect
But we **will** cover...

- What the latest endpoint surveys tell us & what Splunk has seen recently
- Alternatives to the UF, and Best Practices for commercial solutions
- What NOT to do when you collect with the UF
- New Stuff in Sysmon, Windows 10, etc...
- Endpoint Diet! Clever Event Reduction techniques
- An new way to guide which WinEvents to collect

...and lots of hands-on fun with BOTS data in-between!
ENDPOINTS!

LAPTOPS
TOO MUCH %*$&# DATA
DID MY REGISTRY CHANGE?
BREACHES BREACHES BREACHES

NEXT-GEN++++++
Many device types are connecting to networks: desktop computers, followed by employer-owned laptops, network devices and servers, mobile devices, even cloud-based systems, IoT devices, mobile and network devices, and wearables.

What’s an endpoint?  
(courtesy McAfee)
In 2016, we said…the endpoint was important!

- Closest to humans
- Underprotected
- Versatile
- Data-rich
In 2016, we said...the endpoint was important!

The weak link

70% of successful breaches start on the endpoint*

*IDC study 2016

Closest to humans

Underprotected

Versatile

Data-rich
And in 2018, that went up to....
2018: The Endpoint Is STILL Important!
And STILL the weak link

82% of successful breaches involved an endpoint*

*SANS 2018 Endpoint Security Survey

Closest to humans
Underprotected
Versatile
Data-rich
OK, 2019?
OK, 2019?

(j/k...the survey hasn’t been completed yet....)
42% of IT professionals said they had suffered a breach on their endpoints.
20% said they did not know if they had been breached.
82% of those that knew of a breach said it had involved a desktop.
69% cited corporate laptops as the target.
42% cited employee-owned laptops.
Only 47% of antivirus capabilities detected threats.
26% were detected by endpoint detection and response (EDR) capabilities.
For those exploited endpoints, the top threat vectors were found to be web “drive-bys” (63%), social engineering and phishing attacks (53%), and ransomware (50%).
Of the IT professionals that had acquired next-gen endpoint security solutions, 37% haven’t implemented their full capabilities.
49% of those next-gen security solutions possess fileless malware detection features, but 38% of IT professionals haven’t implemented them.
42% of IT professionals said they had suffered a breach on their endpoints.

20% said they did not know if they had been breached.

82% of those that knew of a breach said it had involved a desktop.

69% cited corporate laptops as the target.

42% cited employee-owned laptops.

Only 47% of antivirus capabilities detected threats.

26% were detected by endpoint detection and response (EDR) capabilities.

For those exploited endpoints, the top threat vectors were found to be web “drive-bys” (63%), social engineering and phishing attacks (53%), and ransomware (50%).

50% had purchased a “next-gen” endpoint security solution, and...

Of the IT professionals that had acquired next-gen endpoint security solutions, 37% haven’t implemented their full capabilities.

49% of those next-gen security solutions possess fileless malware detection features, but 38% of IT professionals haven’t implemented them.
AlienVault-Sponsored 2019 Survey

How has endpoint security risk to your organization changed in the last 12 months?

- **53%** See an increase or significant increase in endpoint security risk
- **36%** Significantly increased
- **32%** Significantly decreased
- **17%** No change in importance
- **11%** Becoming less important
- **4%** Becoming more important

How is the importance of endpoint security changing as part of your organization's overall IT security strategy?

- **76%** Becoming more important
- **23%** No change in importance
- **1%** Becoming less important

Splunk Security Specialists:

~5x increase in endpoint assistance requests in 2019
## AlienVault-Sponsored 2019 Survey

<table>
<thead>
<tr>
<th>Impact</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss of end user productivity</td>
<td>52%</td>
</tr>
<tr>
<td>Loss of IT productivity</td>
<td>40%</td>
</tr>
<tr>
<td>System downtime</td>
<td>37%</td>
</tr>
<tr>
<td>Network connections</td>
<td>93%</td>
</tr>
<tr>
<td>File modifications</td>
<td>81%</td>
</tr>
<tr>
<td>Registry changes</td>
<td>74%</td>
</tr>
<tr>
<td>Reputation and brand damage</td>
<td>36%</td>
</tr>
<tr>
<td>Theft of information assets</td>
<td>35%</td>
</tr>
<tr>
<td>Business/revenue impact</td>
<td>33%</td>
</tr>
<tr>
<td>Process information</td>
<td>71%</td>
</tr>
<tr>
<td>Memory content and structures</td>
<td>55%</td>
</tr>
<tr>
<td>User information</td>
<td>50%</td>
</tr>
</tbody>
</table>

**Splunk Security Specialists:** This matches up with the requests that we service!
<table>
<thead>
<tr>
<th>Action</th>
<th>Asset</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hacking - Use of stolen creds</td>
<td>Server - Mail</td>
<td>340</td>
</tr>
<tr>
<td>Social - Phishing</td>
<td>Server - Mail</td>
<td>270</td>
</tr>
<tr>
<td>Social - Phishing</td>
<td>User Dev - Desktop</td>
<td>251</td>
</tr>
<tr>
<td>Malware - Backdoor</td>
<td>User Dev - Desktop</td>
<td>229</td>
</tr>
<tr>
<td>Malware - C2</td>
<td>User Dev - Desktop</td>
<td>210</td>
</tr>
<tr>
<td>Hacking - Use of backdoor or C2</td>
<td>User Dev - Desktop</td>
<td>208</td>
</tr>
<tr>
<td>Malware - Spyware/Keylogger</td>
<td>User Dev - Desktop</td>
<td>103</td>
</tr>
<tr>
<td>Malware - Adminware</td>
<td>User Dev - Desktop</td>
<td>91</td>
</tr>
<tr>
<td>Misuse - Privilege abuse</td>
<td>Server - Database</td>
<td>90</td>
</tr>
<tr>
<td>Malware - Capture app data</td>
<td>Server - Web application</td>
<td>83</td>
</tr>
</tbody>
</table>

**Table 1**
Top action and asset variety combinations within breaches, (n= 2,013)
what about…
• Windows Event Logs: 46% (#1 source by volume)
  • UNIX TA: 16%
  • Windows Perfmon: 6%
  • Windows Registry: 6%
    • McAfee EPO: 6%
  • Symantec Endpoint: 4%
  • Non-Microsoft DNS: 4%
    • Carbon Black: 2%
    • Crowdstrike: 2%
  • Microsoft Sysmon: 1%

(Q1 2019 internal data)
Cisco CSIRT...

Weaponizing Detection

(Valites/Bollinger, 2019)
SANS 2018: Which endpoints and how?

<table>
<thead>
<tr>
<th>Over the past 12 months, what types of endpoints have been compromised?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Please indicate if these were widespread or limited in scope to either a small number of endpoints or just one endpoint. Leave blank all types that were not compromised.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Endpoint Type</th>
<th>Widespread</th>
<th>Small Number of Endpoints</th>
<th>Single Endpoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desktops (employer owned)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laptops (employer owned)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laptops (employee owned)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Servers (line of business applications, legacy)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Servers (development, database, email, web, DNS)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mobile devices (employer owned; tablets, notebooks, PDAs, smartphones)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cloud-based servers (PAAS, emulated or virtualized)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mobile devices (employee owned; tablets, notebooks, PDAs, smartphones)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cloud-based applications (SaaS)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Industrial control systems</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCADA, plant floor manufacturing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Routers/Firewalls/Switches/Other network devices</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Electronic perimeter security systems</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internet of Things devices/Sensors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Environmental controls (HVAC, water treatment)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Printers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Point of sale (POS) devices</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smart systems (cars, building controllers)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smart speakers (Amazon, Google, Echo)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smart sensors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wearables</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>How were these endpoints exploited? Select all that apply.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit_by_bit_hack_or_planting_spyware_to_steal_data</td>
</tr>
<tr>
<td>Ransomware</td>
</tr>
<tr>
<td>Credential theft or compromise</td>
</tr>
<tr>
<td>Exploitation of known vulnerability in published CVE</td>
</tr>
<tr>
<td>Infection of other attackers' USB media device</td>
</tr>
<tr>
<td>Compromised app on the endpoint</td>
</tr>
<tr>
<td>Exploitation of zero day exploit in published CVE</td>
</tr>
<tr>
<td>Exploitation of zero day in published CVE</td>
</tr>
<tr>
<td>Exploitation of zero day in unpatched device</td>
</tr>
<tr>
<td>File Integrity Manipulation</td>
</tr>
<tr>
<td>Machine Interface Vulnerability</td>
</tr>
</tbody>
</table>

Neely, 2018
SANS 2018: Which endpoints and how?

1. Employer-owned Desktops
2. Employer-owned Laptops
3. Employee-owned Laptops
4. Servers (LoB, Legacy)

Let’s get hands-on!
LOGIN INSTRUCTIONS:

Obtain kopek.

Obtain scratch card.

Use kopek to remove the special grey latex ink circle.

Insert number into URL.

Keep kopek for good luck!
Introduction

Splunking the Endpoint V: 2019 Hands-On!

This app showcases tracking the Violent Femmes APT group as they move through the kill chain to eventually exfiltrate data from Frothly, the small homebrewing supply company at the center of (and willing victim within) Splunk's Boss of the SOC (BOTS) v2, v3 and v4 blue-team Capture-The-Flag-esque security competitions. Also included is some material surrounding Frothly's penetration test, and around their acquisition of the Thirsty Berner brewery. A summary of the data available for exploration can be viewed below.

Rather than publish all of the instructions and related resources for this session in an app, we have chosen to provide written collateral to help guide you through the data.

How to Use This App

Simply access the companion Google document in the top level of the folder below and use the "Search" link above or here to run the searches and follow along with the session.

Companion Material Downloads

- Supporting Materials on Google Drive

Select additional content to view using the boxes below:

- Data Summary
- Supporting Apps

Next Step

Next: Search!
VIOLENT MEMMES
**SOCIO-POLITICAL AXIS**

- Seeking to obtain high end Western Beers for production in their breweries

**ADVERSARY**

- Nation-state sponsored adversary
- Uses German naming conventions

**CAPABILITIES**

- PowerShell
- Spearphishing
- Domain Fronting
- Ticket Passing

**INFRASTRUCTURE**

- German Based DigitalOcean servers
- Enom Registered DNS

**TECHNICAL AXIS**

- Metasploit
- Credential Dumping (Mimikatz)
- User svc_print for Account Persistence
- Remote Desktop Protocol
- Sshtasks.exe for beacon

**VICTIMS**

Western innovative Brewers and Home Brewing companies

---

Thanks ThreatConnect!
Hands On! Sysmon and Windows Event Logs....
What was the initial access mechanism into Thirsty Berner for Violent Memmes?

Source types: Microsoft Sysmon and Powershell logging

MITRE ATT&CK: Initial Access
T1192 Spearphishing Link
T1086 Powershell
Several actions occurred when a malicious file that originated with the phishing email was executed. One action resulted in the downloading of a script from a web site. What is the name of the script?

(29 correct!)
(Hands On Redacted)
”This is amazing. I will go back and collect ALL of my Powershell logs!”
A Cautionary Tale
How to get data in...
And avoid trouble doing so!
A guy walks into a Splunk meeting…
This innocent looking `inputs.conf`...
Many for-each statements for iteration = many, many, many log entries in Powershell logs due to use of Microsoft APIs

How many logs?

+ more below…
Teh badness.
~2000 per run.
<table>
<thead>
<tr>
<th>_time</th>
<th>1</th>
<th>4103</th>
<th>4104</th>
<th>4688</th>
<th>500</th>
<th>501</th>
<th>800</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-09-06 00:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 01:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 02:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 03:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 04:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 05:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 06:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 07:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 08:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 09:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 10:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 11:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 12:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 13:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 14:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 15:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 16:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 17:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 18:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 19:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 20:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 21:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 22:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 23:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
</tbody>
</table>

0.01 | 10.35 | 29.54 | 0.05 | 8.04 | 8.14 | 9.39 |
<table>
<thead>
<tr>
<th>_time</th>
<th>1</th>
<th>4103</th>
<th>4104</th>
<th>4688</th>
<th>500</th>
<th>501</th>
<th>800</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-09-06 00:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 01:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 02:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 03:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 04:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 05:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 06:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 07:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 08:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 09:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 10:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 11:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 12:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 13:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 14:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 15:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 16:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 17:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 18:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 19:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 20:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 21:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 22:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 23:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
</tbody>
</table>

~56 MB Per day per host from ONE POWERSHELL SCRIPT.
let’s do the math

56/24 = 2.3MB per hour
2.3MB * 10 hours daily = 23MB per endpoint
23 * 16,000 = 368GB a day

...except ~1/3rd were servers, so...

23 * 11,000 = 253GB  and  56 * 5,000 = 280GB

533GB a day.
<table>
<thead>
<tr>
<th>_time</th>
<th>1</th>
<th>4103</th>
<th>4104</th>
<th>4688</th>
<th>500</th>
<th>501</th>
<th>800</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-09-06 00:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 01:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 02:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 03:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 04:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 05:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 06:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 07:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 08:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 09:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 10:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 11:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 12:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 13:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 14:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 15:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 16:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 17:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 18:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 19:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 20:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 21:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 22:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
<tr>
<td>2019-09-06 23:00</td>
<td>0.00</td>
<td>0.43</td>
<td>1.23</td>
<td>0.00</td>
<td>0.33</td>
<td>0.34</td>
<td>0.39</td>
</tr>
</tbody>
</table>

OMG.
What are they? Should we collect?

4104 = Almost always yes
4103 = Sometimes...
800 = same as 4103!
50x = ”largely useless” – basically logs starts and stops
4100 = Sure, minor volume

What’s interesting to note is that newer versions of PowerShell will often log to both event logs simultaneously.

https://www.eventsentry.com/blog/2018/01/powershell-p0wrh11-securing-powershell.html
Three places to get example blacklists...

1. Version 6.0 of the Windows TA (Splunkbase)

2. Automine’s (David Shpritz)’s Github and related presentation:
   https://gist.github.com/automine/a3915d5238e2967c8d44b0ebcfb66147

3. What we used for BOTS
   https://splk.it/conf19-splunk-endpoint
Here’s where we ended up for SecKit IDM…

[WinEventLog://Microsoft-Windows-Powershell/Operational]
index = main
disabled = false
renderXml = 0
blacklist = EventCode="4104" Message="(?::Path:).+(?:\s\s)\s.*splunk-powershell-common.ps1"
blacklist1 = EventCode="4104" Message="(?::Path:).+(?:\s\s)\s.*splunk-powershell.ps1"
blacklist2 = EventCode="4104" Message="(?::Path:).+(?:\s\s)\s.*generate_windows_update_logs.ps1"
blacklist3 = EventCode="4103" Message="(?:Host Application = ).(?:.*splunk-powershell.ps1\s.*)"
blacklist4 = EventCode="(4104|4103)" Message="(?::Path:).+(?:\s\s)\s.*get-AllInterfaceConfig.ps1"
blacklist5 = EventCode="4103" Message="(?:Host Application = ).(?:.*\s\s)\sget-AllInterfaceConfig.ps1"

[WinEventLog://Windows PowerShell]
index = main
disabled = false
renderXml = 0
blacklist = EventCode="(800|500|501)" Message="(?::HostApplication=).(?:.*\s\s)\sget-AllInterfaceConfig.ps1"
### EventCode

**3 Values, 100% of events**

<table>
<thead>
<tr>
<th>Reports</th>
<th>Average over time</th>
<th>Maximum value over time</th>
<th>Minimum value over time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top values</td>
<td>Top values by time</td>
<td>Rare values</td>
<td></td>
</tr>
<tr>
<td>Events with this field</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Avg:** 2733.7389704716293  **Min:** 800  **Max:** 4104  **Std Dev:** 1627.403367599553

<table>
<thead>
<tr>
<th>Values</th>
<th>Count</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>4103</td>
<td>13,072</td>
<td>41.46%</td>
</tr>
<tr>
<td>808</td>
<td>13,072</td>
<td>41.46%</td>
</tr>
<tr>
<td>4104</td>
<td>5,385</td>
<td>17.08%</td>
</tr>
</tbody>
</table>

### EventCode

**2 Values, 100% of events**

<table>
<thead>
<tr>
<th>Reports</th>
<th>Average over time</th>
<th>Maximum value over time</th>
<th>Minimum value over time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top values</td>
<td>Top values by time</td>
<td>Rare values</td>
<td></td>
</tr>
<tr>
<td>Events with this field</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Avg:** 4103.9977974888619  **Min:** 4103  **Max:** 4104  **Std Dev:** 0.04496476106941514

<table>
<thead>
<tr>
<th>Values</th>
<th>Count</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>4104</td>
<td>2,464</td>
<td>99.797%</td>
</tr>
<tr>
<td>4103</td>
<td>5</td>
<td>0.202%</td>
</tr>
</tbody>
</table>
**But … FAIL. It is still 1.2MB per run!**

<table>
<thead>
<tr>
<th><em>time</em></th>
<th>bytes</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-10-17 04:00</td>
<td>135258</td>
<td>1.2962818145751953</td>
</tr>
<tr>
<td>2019-10-17 05:00</td>
<td>1291778</td>
<td>1.2319355010986328</td>
</tr>
<tr>
<td>2019-10-17 06:00</td>
<td>1291778</td>
<td>1.2319355010986328</td>
</tr>
<tr>
<td>2019-10-17 07:00</td>
<td>1289974</td>
<td>1.238215072631836</td>
</tr>
<tr>
<td>2019-10-17 08:00</td>
<td>1290876</td>
<td>1.2310752868652344</td>
</tr>
<tr>
<td>2019-10-17 09:00</td>
<td>1292229</td>
<td>1.232386598215332</td>
</tr>
<tr>
<td>2019-10-17 10:00</td>
<td>1292229</td>
<td>1.232386598215332</td>
</tr>
<tr>
<td>2019-10-17 11:00</td>
<td>129523</td>
<td>1.2297849655151367</td>
</tr>
<tr>
<td>2019-10-17 12:00</td>
<td>1298876</td>
<td>1.2318752868552344</td>
</tr>
<tr>
<td>2019-10-17 13:00</td>
<td>1291778</td>
<td>1.2319355010986328</td>
</tr>
<tr>
<td>2019-10-17 14:00</td>
<td>1291327</td>
<td>1.231505399819336</td>
</tr>
<tr>
<td>2019-10-17 15:00</td>
<td>1291778</td>
<td>1.2319355010986328</td>
</tr>
<tr>
<td>2019-10-17 16:00</td>
<td>1292229</td>
<td>1.2236598215332</td>
</tr>
<tr>
<td>2019-10-17 17:00</td>
<td>1291327</td>
<td>1.231505399819336</td>
</tr>
<tr>
<td>2019-10-17 18:00</td>
<td>1291327</td>
<td>1.231505399819336</td>
</tr>
<tr>
<td>2019-10-17 19:00</td>
<td>1294833</td>
<td>1.234086036682129</td>
</tr>
<tr>
<td>2019-10-17 20:00</td>
<td>1298425</td>
<td>1.2386451797485352</td>
</tr>
<tr>
<td>2019-10-17 21:00</td>
<td>1292229</td>
<td>1.232386598215332</td>
</tr>
</tbody>
</table>

**Because you can’t filter the 4104...**
But 4104 events “automatically” warn for suspicious modules? Maybe only collect those?
Nope.
Almost 200 modules are “Warning” worthy.

LESSONS LEARNED!

• The SecKit IDM Interface Config powershell script is fundamentally incompatible with recommended powershell logging. The 4104 from it are unfilterable at the UF/HF level. Reduce interval?

• Many other useful powershell logs may be difficult to filter: YMMV.

• Make sure you aren’t collecting duplicate info (4103 and 800!)

• Make sure you know what you’re collecting, at what interval, and why! Maybe an alternative to powershell for gathering?
Can we filter better?

YES. But first…
“Thanks for the advice. But what event codes SHOULD we collect?”
What should we collect, and why?

That's a good question!
We typically answer with...
We typically answer with...

What if we had an app for that?
Hands On! Windows Event Code Guidance!
This beta app allows a Splunk admin or security analyst to make better decisions about which Windows Event Codes are most important for traditional security use cases such as security investigation, incident response, and advanced threat hunting. Recommendations from six different security researchers/organizations have been included in the app via a lookup table, encompassing 567 different events, most of which are from the Windows Security event log. Start with the Lookup Overview above to get a feel for the event codes and recommendations, and drill down on any event codes to see the details of that event code in your Splunk instance. You may also interact with your Windows Event Code data in a tabular (Table Analysis) and graphical (Treemap Analysis) format. Finally, you can pick individual hosts and see which Event Codes are being collected from that host, and compare those codes against recommendations and ingest levels.
About This App

This beta app allows a Splunk admin or security analyst to make better decisions about which Windows Event Codes are most important for traditional security use cases such as security investigation, incident response, and advanced threat hunting. Recommendations from six different security researchers/organizations have been included in the app via a lookup table, encompassing 567 different events, most of which are from the Windows Security event log. Start with the Lookup Overview above to get a feel for the event codes and recommendations, and drill down on any event codes to see the details of that event code in your Splunk instance. You may also interact with your Windows Event Code data in a tabular (Table Analysis) and graphical (Treemap Analysis) format. Finally, you can pick individual hosts and see which Event Codes are being collected from that host, and compare those codes against recommendations and ingest levels.
## Lookup Overview

Select one or more Authorities using the filter

**Authority**

- Michael Gough
- NSA

### Current Filter: 2 Authorities

`ec_guidance_gough=1 OR ec_guidance_nsa=1`

### Number of Event Codes Total in Lookup

![567 Event Codes in Lookup](image)

### Number of Event Codes Selected (2 selected)

![220 Event Codes Selected](image)

## Top 10 Event Log Sources (2 selected)

<table>
<thead>
<tr>
<th>Event Log</th>
<th>count</th>
<th>percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>92</td>
<td>41.8182</td>
</tr>
<tr>
<td>Microsoft-Windows-Defender/Operational</td>
<td>20</td>
<td>9.0909</td>
</tr>
<tr>
<td>System</td>
<td>17</td>
<td>7.2727</td>
</tr>
<tr>
<td>Microsoft-Windows-MLAN-AutoConfig/Operational</td>
<td>13</td>
<td>5.9091</td>
</tr>
<tr>
<td>Application</td>
<td>11</td>
<td>5.0000</td>
</tr>
<tr>
<td>Microsoft-Windows-Powershell/Operational</td>
<td>7</td>
<td>3.1818</td>
</tr>
<tr>
<td>System or Sysmon</td>
<td>6</td>
<td>2.7273</td>
</tr>
<tr>
<td>Microsoft-Windows-Application-Experience/Program-Inventory</td>
<td>6</td>
<td>2.7273</td>
</tr>
<tr>
<td>Microsoft-Windows-Taskscheduler/Operational</td>
<td>5</td>
<td>2.2727</td>
</tr>
</tbody>
</table>

### Codes Ranked by Weight (2 selected)

<table>
<thead>
<tr>
<th>EventCode</th>
<th>Event Log</th>
<th>EventDescription</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>4624</td>
<td>Security</td>
<td>An account was successfully logged on.</td>
<td>6</td>
</tr>
<tr>
<td>4625</td>
<td>Security</td>
<td>An account failed to log on.</td>
<td>6</td>
</tr>
<tr>
<td>4657</td>
<td>Security</td>
<td>A registry value was modified.</td>
<td>5</td>
</tr>
<tr>
<td>4719</td>
<td>Security</td>
<td>System audit policy was changed.</td>
<td>5</td>
</tr>
<tr>
<td>5140</td>
<td>Security</td>
<td>A network share object was accessed.</td>
<td>5</td>
</tr>
<tr>
<td>4634</td>
<td>Security</td>
<td>An account was logged off.</td>
<td>4</td>
</tr>
<tr>
<td>4648</td>
<td>Security</td>
<td>A login was attempted using explicit credentials.</td>
<td>4</td>
</tr>
<tr>
<td>4688</td>
<td>Security</td>
<td>A new process has been created.</td>
<td>4</td>
</tr>
<tr>
<td>4720</td>
<td>Security</td>
<td>A user account was created.</td>
<td>4</td>
</tr>
<tr>
<td>4722</td>
<td>Security</td>
<td>A user account was enabled.</td>
<td>4</td>
</tr>
</tbody>
</table>

This table displays, for the current selected authorities, what event codes are recommended from those authorities and what event sources they come from.
## Lookup Overview

Select one or more Authorities using the filter

### Authority

- Michael Gough
- NSA

### Current Filter: 2 Authorities

`ec_guidance_gough=1 OR ec_guidance_nsa=1`

### Number of Event Codes Total in Lookup

<table>
<thead>
<tr>
<th>Event Codes in Lookup</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>567</td>
<td></td>
</tr>
</tbody>
</table>

### Number of Event Codes Selected (2 selected)

<table>
<thead>
<tr>
<th>Event Codes Selected</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>220</td>
<td></td>
</tr>
</tbody>
</table>

---

### Top 10 Event Log Sources (2 selected)

<table>
<thead>
<tr>
<th>Event Log Source</th>
<th>Event Log</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td></td>
<td>92</td>
</tr>
<tr>
<td>Microsoft-Windows Defender/Operational</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>System</td>
<td></td>
<td>17</td>
</tr>
<tr>
<td>Microsoft-Windows-MLAN-AutoConfig/Operational</td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>Application</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>Microsoft-Windows-Powershell/Operational</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>System or System</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>Microsoft-Windows-Application-Experience/Program-Inventory</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>Microsoft-Windows-TaskScheduler/Operational</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Microsoft-Windows-CodeIntegrity/Operational</td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>

This table displays, for the current selected authorities, what event codes are recommended from those authorities and what event sources they come from.

---

### Event Log  (2 selected)

<table>
<thead>
<tr>
<th>Event Code</th>
<th>Event Log</th>
<th>EventDescription</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>4624</td>
<td>Security</td>
<td>An account was successfully logged on.</td>
<td>6</td>
</tr>
<tr>
<td>4625</td>
<td>Security</td>
<td>An account failed to log on.</td>
<td>6</td>
</tr>
<tr>
<td>4657</td>
<td>Security</td>
<td>A registry value was modified.</td>
<td>5</td>
</tr>
<tr>
<td>4719</td>
<td>Security</td>
<td>System audit policy was changed.</td>
<td>5</td>
</tr>
<tr>
<td>4720</td>
<td>Security</td>
<td>A network share object was accessed.</td>
<td>5</td>
</tr>
<tr>
<td>4721</td>
<td>Security</td>
<td>An account was logged off.</td>
<td>5</td>
</tr>
<tr>
<td>4648</td>
<td>Security</td>
<td>A logon was attempted using explicit credentials.</td>
<td>4</td>
</tr>
<tr>
<td>4688</td>
<td>Security</td>
<td>A new process has been created.</td>
<td>4</td>
</tr>
<tr>
<td>4720</td>
<td>Security</td>
<td>A user account was created.</td>
<td>4</td>
</tr>
<tr>
<td>4722</td>
<td>Security</td>
<td>A user account was enabled.</td>
<td>4</td>
</tr>
</tbody>
</table>

This table displays, for the current selected authorities, what event codes are recommended from those authorities and how many sources (in total) suggest that event code should be collected.
### Count of Codes by Authority (2 selected)

<table>
<thead>
<tr>
<th>Category</th>
<th>Total EventCodes</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Michael Gough</td>
<td>49</td>
<td><a href="https://www.malwarearchaeology.com/cheat-sheets">https://www.malwarearchaeology.com/cheat-sheets</a></td>
</tr>
</tbody>
</table>

### Michael Gough ATT&CK Mapping (2 selected)


This table displays, for the currently selected authorities, what overlap exists with other authorities. In other words, “for my currently selected authorities, what other authorities recommend how many of the same event codes?"
SCROLL UP
Click on 4688.
## Individual Event Code Analysis

### Sources
- **Aug 2019**
- **ALL**

### Indexes
- **ALL**

### Event Code
- **4688**

### Source Type
- **wineventlog**
- **xmlwineventlog**

### Graph
- Time series chart showing data from Thu Aug 1 2019 to Thu Aug 29 2019.

### Table
<table>
<thead>
<tr>
<th>Event Code</th>
<th>Event Log</th>
<th>Event Description</th>
<th>Number of Recommendations</th>
<th>Source Type</th>
<th>Number of Hosts</th>
<th>Number of Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>4688</td>
<td>Security</td>
<td>A new process has been created.</td>
<td>4</td>
<td>WinEventLog</td>
<td>10</td>
<td>11600</td>
</tr>
</tbody>
</table>

### Analysis
- **10** Hosts with this Event Code
- **YES** Tagged somewhere in CIM?
- **22** MB seen from this event in time selected
- **2** Avg MB seen per host in time selected
- **YES** MITRE Attack Framework?
- **NO** Possible Duplicate?

### Recommendations
- Andrea Fortuna
- Michael Gough
- Mike Lombardi
- NSA
- SANS Forensics Guidance

- **Recommends?**
  - NO
  - YES
  - YES
  - YES
  - NO
Click and select “Recommended Events Table”
1. Select “ALL” and “main”

2. Click!
Which events are we collecting that we “should” be, and from how many hosts?
Which events are we collecting that we MAYBE should NOT (for security use cases), and from how many hosts?

<table>
<thead>
<tr>
<th>EventCode</th>
<th>ATT&amp;CK</th>
<th>Category</th>
<th>Event Log</th>
<th>EventDescription</th>
<th>Level</th>
<th>NumHosts</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>0</td>
<td>Sysmon</td>
<td>File Create Stream Hash</td>
<td>Information</td>
<td>12</td>
<td>WinEventLog:Application, WinEventLog:System</td>
<td>Sysmon</td>
</tr>
<tr>
<td>10016</td>
<td>Not in Lookup</td>
<td>Not in Lookup</td>
<td>Not in Lookup</td>
<td>Not in Lookup</td>
<td>10</td>
<td>WinEventLog:System</td>
<td>Not in Lookup</td>
</tr>
<tr>
<td>10083</td>
<td>Not in Lookup</td>
<td>Not in Lookup</td>
<td>Not in Lookup</td>
<td>Not in Lookup</td>
<td>10</td>
<td>WinEventLog:Application, WinEventLog:System</td>
<td>Not in Lookup</td>
</tr>
<tr>
<td>16394</td>
<td>Not in Lookup</td>
<td>Not in Lookup</td>
<td>Not in Lookup</td>
<td>Not in Lookup</td>
<td>10</td>
<td>WinEventLog:Application, WinEventLog:System</td>
<td>Not in Lookup</td>
</tr>
<tr>
<td>8198</td>
<td>Not in Lookup</td>
<td>Not in Lookup</td>
<td>Not in Lookup</td>
<td>Not in Lookup</td>
<td>10</td>
<td>WinEventLog:Application, WinEventLog:System</td>
<td>Not in Lookup</td>
</tr>
</tbody>
</table>
Same question, answered graphically…
<table>
<thead>
<tr>
<th>EventCode</th>
<th>ATT&amp;CK</th>
<th>Category</th>
<th>Event Log</th>
<th>EventDescription</th>
<th>Event Type</th>
<th>Source</th>
<th>Subcategory</th>
<th>duplicate_possible</th>
<th>observed_volume</th>
<th>NumRecommenders</th>
</tr>
</thead>
<tbody>
<tr>
<td>4624</td>
<td>1</td>
<td>Logon/Logoff</td>
<td>Security</td>
<td>An account was logged on.</td>
<td>Logon</td>
<td>Information</td>
<td>WinEventLog:Security</td>
<td>0</td>
<td>In Development</td>
<td>7</td>
</tr>
<tr>
<td>4634</td>
<td>0</td>
<td>Logon/Logoff</td>
<td>Security</td>
<td>An account was logged off.</td>
<td>Logoff</td>
<td>Information</td>
<td>WinEventLog:Security</td>
<td>0</td>
<td>In Development</td>
<td>5</td>
</tr>
<tr>
<td>4648</td>
<td>0</td>
<td>Logon/Logoff</td>
<td>Security</td>
<td>A logon was attempted using explicit credentials.</td>
<td>Logon</td>
<td>Information</td>
<td>WinEventLog:Security</td>
<td>0</td>
<td>In Development</td>
<td>5</td>
</tr>
<tr>
<td>4672</td>
<td>0</td>
<td>Privilege Use</td>
<td>Security</td>
<td>Special privileges assigned to new logon.</td>
<td>Privilege Use</td>
<td>Information</td>
<td>WinEventLog:Security</td>
<td>Sensitive Privilege Use / Non Sensitive Privilege Use</td>
<td>0</td>
<td>In Development</td>
</tr>
<tr>
<td>4688</td>
<td>1</td>
<td>Detailed Tracking</td>
<td>Security</td>
<td>A new process has been created.</td>
<td>Process Creation</td>
<td>Information</td>
<td>WinEventLog:Security</td>
<td>0</td>
<td>In Development</td>
<td>5</td>
</tr>
<tr>
<td>4647</td>
<td>0</td>
<td>Logon/Logoff</td>
<td>Security</td>
<td>User initiated logoff.</td>
<td>Logoff</td>
<td>Information</td>
<td>WinEventLog:Security</td>
<td>0</td>
<td>In Development</td>
<td>4</td>
</tr>
<tr>
<td>4625</td>
<td>0</td>
<td>Logon/Logoff</td>
<td>Security</td>
<td>An account failed to log on.</td>
<td>Logon</td>
<td>Information</td>
<td>WinEventLog:Security</td>
<td>1</td>
<td>In Development</td>
<td>4</td>
</tr>
<tr>
<td>4719</td>
<td>0</td>
<td>Policy Change</td>
<td>Security</td>
<td>System audit policy was changed.</td>
<td>Audit Policy Change</td>
<td>Information</td>
<td>WinEventLog:Security</td>
<td>0</td>
<td>In Development</td>
<td>7</td>
</tr>
<tr>
<td>4778</td>
<td>0</td>
<td>Logon/Logoff</td>
<td>Security</td>
<td>A session was reconnected to a Windows Station.</td>
<td>Other Logon/Logoff Events</td>
<td>Information</td>
<td>WinEventLog:Security</td>
<td>0</td>
<td>In Development</td>
<td>5</td>
</tr>
<tr>
<td>4779</td>
<td>0</td>
<td>Logon/Logoff</td>
<td>Security</td>
<td>A session was disconnected from a Windows Station.</td>
<td>Other Logon/Logoff Events</td>
<td>Information</td>
<td>WinEventLog:Security</td>
<td>0</td>
<td>In Development</td>
<td>5</td>
</tr>
<tr>
<td>7845</td>
<td>1</td>
<td>System</td>
<td>System</td>
<td>New Windows Service</td>
<td>Service</td>
<td>Information</td>
<td>WinEventLog:System</td>
<td>0</td>
<td>In Development</td>
<td>4</td>
</tr>
<tr>
<td>4720</td>
<td>0</td>
<td>Account Management</td>
<td>Security</td>
<td>A user account was created.</td>
<td>User Account Management</td>
<td>Information</td>
<td>WinEventLog:Security</td>
<td>0</td>
<td>In Development</td>
<td>5</td>
</tr>
<tr>
<td>4722</td>
<td>0</td>
<td>Account Management</td>
<td>Security</td>
<td>A user account was enabled.</td>
<td>User Account Management</td>
<td>Information</td>
<td>WinEventLog:Security</td>
<td>0</td>
<td>In Development</td>
<td>5</td>
</tr>
</tbody>
</table>
1. Select "All Time" – NOT REAL TIME ALL TIME

2. Click Submit!
Onboard logs from your “golden image” and analyze!
Where do I get it?

1. From the link provided in the Endpoint App:
   
   https://splk.it/conf19-splunk-endpoint

2. Github:
   
   https://github.com/stressboi/splunk_wineventcode_secanalysis

COMING! jp-CERT analysis as a 7th source!
Enabling process auditing and sending all the endpoint event logs to Splunk
Even with the best intentions...

Splunk eats too much.
What’s normal?

- MEDIUM
- BIGGIE
- GREAT BIGGIE
- SMALL
- MEDIUM
- LARGE
“ALL DATA IS SECURITY RELEVANT.”

--many people at Splunk, many times.
What kind of endpoints and how?

1. Employer-owned Desktops
2. Employer-owned Laptops
3. Employee-owned Laptops
4. Servers (LoB, Legacy)

If these are the most likely endpoints to be compromised, shouldn’t we be collecting logs from them?
What to collect from user endpoints?

Using the Universal Forwarder on Windows

► Basic
  • Windows Event logs
  • Security
    – Set up command process auditing (4688)
  • System
  • Application
  • WindowsUpdateLog (on supported systems)

► Intermediate
  • Sysmon (with TaySwift or Olaf config + Splunk Tweaks)
  • Captures registry instead of Splunk regmon
  • Powershell
    • Module Logging
    • Script Block Logging
    • Scripted Inputs

► Advanced/Specific
  • Splunk Stream
  • Perfmon
  • Powershell Transcription Logs
  • Applocker
  • Windows Firewall
  • WinPrintMon
  • Native USB Auditing
What to collect from user endpoints?
Using the Universal Forwarder on Windows

▶ Basic
• Windows Event logs
• Security
  - Set up command process auditing (4688)
• System
• Application
• WindowsUpdateLog (on supported systems)

▶ Intermediate
• Sysmon (with TaySwift or Olaf config + Splunk Tweaks)
• Captures registry instead of Splunk regmon
• Powershell
• Module Logging
• Script Block Logging
• Scripted Inputs

▶ Advanced/Specific
• Splunk Stream
• Perfmon
• Powershell Transcription Logs
• Applocker
• Windows Firewall
• WinPrintMon
• Native USB Auditing

And what happens if you collect absolutely everything with no filtering?
Storage and compute doesn’t grow on trees.
Today, client-side attacks are more common

- Means the attack occurs at the desktop
- Which means you need desktop logs...

Yet, cost of desktop logs is considered too high

- If strategy is collect everything, that is true
- If strategy is to stay nimble and tactical, it is more expensive not to log...

Advanced agent filtering is helpful or file server tricks
What did we collect this year for BOTS?

• Latest UF (7.3.x) on every endpoint
• Latest Windows TA with all standard scripted inputs enabled except none of the “Mon” inputs (regmon, netmon, printmon, etc)
• Windows Security, System, Application Events using Michael Gough’s audit config and some blacklisting on Security events
• Microsoft Sysmon v10 with Olaf Hartong’s latest config + some more Splunk filtering tweaks
• Windows Powershell/Operational log (4103 and 4104 events)
• CB Response with watchlists and five standard threat feeds, as well as netconn and process events
• Splunk Stream collecting DNS, HTTP, TCP, UDP, DHCP and a few other protocols

To gauge ingest levels we look at Windows Events, Sysmon, Scripted TA output, and Powershell.
Upwards of 50MB per endpoint? Uhoh.
In general, we had lots of extra stuff.

4673 = Not recommended to collect

4663 = Granular object auditing

4688 = Critical, but can also use Sysmon 1

4689 = Not recommended to collect
If we remove those four codes...

<table>
<thead>
<tr>
<th>Time</th>
<th>ABUNGSTEIN_L0</th>
<th>AGRADYL_0</th>
<th>BISTOLL_L0</th>
<th>BTUN_L0</th>
<th>FMalteKeskio_l0</th>
<th>GHOPPY_L0</th>
<th>JWortoski_L0</th>
<th>MVALITUS_L0</th>
<th>PCERF_L0</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011-08-02 00:00</td>
<td>0.064</td>
<td>0.248</td>
<td>2.728</td>
<td>5.525</td>
<td>0.219</td>
<td>0.423</td>
<td>1.451</td>
<td>0.382</td>
<td>0.526</td>
</tr>
<tr>
<td>2011-08-02 01:00</td>
<td>0.215</td>
<td>0.283</td>
<td>0.334</td>
<td>0.187</td>
<td>0.333</td>
<td>1.129</td>
<td>0.157</td>
<td>0.389</td>
<td>0.299</td>
</tr>
<tr>
<td>2011-08-02 02:00</td>
<td>0.008</td>
<td>0.181</td>
<td>0.285</td>
<td>0.851</td>
<td>0.011</td>
<td>0.273</td>
<td>0.085</td>
<td>0.377</td>
<td>0.296</td>
</tr>
<tr>
<td>2011-08-02 03:00</td>
<td>0.028</td>
<td>0.179</td>
<td>0.067</td>
<td>0.038</td>
<td>0.015</td>
<td>0.060</td>
<td>0.062</td>
<td>0.085</td>
<td>0.382</td>
</tr>
<tr>
<td>2011-08-02 04:00</td>
<td>0.064</td>
<td>0.248</td>
<td>3.738</td>
<td>5.279</td>
<td>0.219</td>
<td>0.433</td>
<td>1.451</td>
<td>0.382</td>
<td>0.526</td>
</tr>
<tr>
<td>2011-08-02 05:00</td>
<td>0.215</td>
<td>0.283</td>
<td>0.334</td>
<td>0.187</td>
<td>0.333</td>
<td>1.129</td>
<td>0.157</td>
<td>0.389</td>
<td>0.299</td>
</tr>
<tr>
<td>2011-08-02 06:00</td>
<td>0.008</td>
<td>0.181</td>
<td>0.285</td>
<td>0.851</td>
<td>0.011</td>
<td>0.273</td>
<td>0.085</td>
<td>0.377</td>
<td>0.296</td>
</tr>
<tr>
<td>2011-08-02 07:00</td>
<td>0.028</td>
<td>0.179</td>
<td>0.067</td>
<td>0.038</td>
<td>0.015</td>
<td>0.060</td>
<td>0.062</td>
<td>0.085</td>
<td>0.382</td>
</tr>
<tr>
<td>2011-08-02 08:00</td>
<td>0.064</td>
<td>0.248</td>
<td>2.728</td>
<td>5.525</td>
<td>0.219</td>
<td>0.423</td>
<td>1.451</td>
<td>0.382</td>
<td>0.526</td>
</tr>
<tr>
<td>2011-08-02 09:00</td>
<td>0.215</td>
<td>0.283</td>
<td>0.334</td>
<td>0.187</td>
<td>0.333</td>
<td>1.129</td>
<td>0.157</td>
<td>0.389</td>
<td>0.299</td>
</tr>
<tr>
<td>2011-08-02 10:00</td>
<td>0.008</td>
<td>0.181</td>
<td>0.285</td>
<td>0.851</td>
<td>0.011</td>
<td>0.273</td>
<td>0.085</td>
<td>0.377</td>
<td>0.296</td>
</tr>
<tr>
<td>2011-08-02 11:00</td>
<td>0.028</td>
<td>0.179</td>
<td>0.067</td>
<td>0.038</td>
<td>0.015</td>
<td>0.060</td>
<td>0.062</td>
<td>0.085</td>
<td>0.382</td>
</tr>
<tr>
<td>2011-08-02 12:00</td>
<td>0.064</td>
<td>0.248</td>
<td>2.728</td>
<td>5.525</td>
<td>0.219</td>
<td>0.433</td>
<td>1.451</td>
<td>0.382</td>
<td>0.526</td>
</tr>
<tr>
<td>2011-08-02 13:00</td>
<td>0.215</td>
<td>0.283</td>
<td>0.334</td>
<td>0.187</td>
<td>0.333</td>
<td>1.129</td>
<td>0.157</td>
<td>0.389</td>
<td>0.299</td>
</tr>
<tr>
<td>2011-08-02 14:00</td>
<td>0.008</td>
<td>0.181</td>
<td>0.285</td>
<td>0.851</td>
<td>0.011</td>
<td>0.273</td>
<td>0.085</td>
<td>0.377</td>
<td>0.296</td>
</tr>
<tr>
<td>2011-08-02 15:00</td>
<td>0.028</td>
<td>0.179</td>
<td>0.067</td>
<td>0.038</td>
<td>0.015</td>
<td>0.060</td>
<td>0.062</td>
<td>0.085</td>
<td>0.382</td>
</tr>
<tr>
<td>2011-08-02 16:00</td>
<td>0.064</td>
<td>0.248</td>
<td>2.728</td>
<td>5.525</td>
<td>0.219</td>
<td>0.433</td>
<td>1.451</td>
<td>0.382</td>
<td>0.526</td>
</tr>
<tr>
<td>2011-08-02 17:00</td>
<td>0.215</td>
<td>0.283</td>
<td>0.334</td>
<td>0.187</td>
<td>0.333</td>
<td>1.129</td>
<td>0.157</td>
<td>0.389</td>
<td>0.299</td>
</tr>
<tr>
<td>2011-08-02 18:00</td>
<td>0.008</td>
<td>0.181</td>
<td>0.285</td>
<td>0.851</td>
<td>0.011</td>
<td>0.273</td>
<td>0.085</td>
<td>0.377</td>
<td>0.296</td>
</tr>
<tr>
<td>2011-08-02 19:00</td>
<td>0.028</td>
<td>0.179</td>
<td>0.067</td>
<td>0.038</td>
<td>0.015</td>
<td>0.060</td>
<td>0.062</td>
<td>0.085</td>
<td>0.382</td>
</tr>
<tr>
<td>2011-08-02 20:00</td>
<td>0.064</td>
<td>0.248</td>
<td>2.728</td>
<td>5.525</td>
<td>0.219</td>
<td>0.433</td>
<td>1.451</td>
<td>0.382</td>
<td>0.526</td>
</tr>
<tr>
<td>2011-08-02 21:00</td>
<td>0.215</td>
<td>0.283</td>
<td>0.334</td>
<td>0.187</td>
<td>0.333</td>
<td>1.129</td>
<td>0.157</td>
<td>0.389</td>
<td>0.299</td>
</tr>
<tr>
<td>2011-08-02 22:00</td>
<td>0.008</td>
<td>0.181</td>
<td>0.285</td>
<td>0.851</td>
<td>0.011</td>
<td>0.273</td>
<td>0.085</td>
<td>0.377</td>
<td>0.296</td>
</tr>
<tr>
<td>2011-08-02 23:00</td>
<td>0.028</td>
<td>0.179</td>
<td>0.067</td>
<td>0.038</td>
<td>0.015</td>
<td>0.060</td>
<td>0.062</td>
<td>0.085</td>
<td>0.382</td>
</tr>
</tbody>
</table>

Best case, ~6MB a day, worst, ~12MB!
BOTS Lessons Learned

1. If you can at all use Sysmon, do so. Much more granular and flexible filtering for process events, file creates. 4688 is better than nothing.

2. Be ruthless about what event codes you collect. Collect the ones that meet your use case and are “recommended.”

3. renderXML=true may save you some space, we used Classic because of some issues we found with blacklisting...
- Large, Fortune 500 company based in the US
- 70,000 Windows endpoints running Carbon Black Response
- `cb-event-forwarder` to get raw sensor data in Splunk
- COLLECT: Process info, network connection info, alerts, watchlists
- NOT COLLECT: File modifications, registry modifications, and module loads: diminishing returns from both splunk license and storage perspective…

(and if you need to, you can always hunt this stuff in the native tool.)

600GB a day (about 8.5MB per endpoint, per day!)
SEC1952 - Finding Evil Is Never An Accident: How to Hunt in BOTS

**SPEAKERS**

Michael Haag, Director of Advanced Threat Detection, Red Canary

To secure the modern endpoint, you need sufficient data, the right visibility and analysis, and the technology necessary to stop an intrusion. We will leverage BOTSv4 data in this session to help you test and validate Splunk use cases related to...

**Industries:** Not industry specific  
**Products:** Splunk Enterprise, Splunk Cloud
What our BOTS machines collected from CB

Event Collection

Disabling event collection will impact visibility, but may improve sensor and server performance.

Process Events
- Process Information
  Collect metadata including starts, stops, pid.
- Process user context
  Collect username associated with events.
- File modifications
  Record modifications of binary files, eg. dll/exe.
    - Non-binary file writes
      Record filemod events for non-binary files.
- Binary module loads
  Collect load events for .dll, .sys, .exe, .so, .dylib.
- Network connections
  Collect in/outgoing network events.

Windows Events
- Cross process events
  Collect events across process boundaries.
- Registry modifications
  Collect write and delete events in the registry.
- EMET events
  Collect EMET mitigation and protection events.

Binary / Module / Storefile Events
- Binaries
  Collect binary modules.
- Binary info
  Collect metadata that describes binaries.
What our BOTS machines looked like from CB

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-08-02 00:00</td>
<td>0.542</td>
<td>0.948</td>
<td>2.885</td>
<td>2.894</td>
<td>1.317</td>
<td>1.758</td>
<td>4.967</td>
<td>0.739</td>
<td>0.866</td>
</tr>
<tr>
<td>2019-08-02 01:00</td>
<td>0.587</td>
<td>0.659</td>
<td>0.616</td>
<td>0.824</td>
<td>0.243</td>
<td>1.352</td>
<td>4.582</td>
<td>0.460</td>
<td>0.849</td>
</tr>
<tr>
<td>2019-08-02 02:00</td>
<td>0.366</td>
<td>0.611</td>
<td>0.663</td>
<td>0.492</td>
<td>1.290</td>
<td>0.494</td>
<td>3.766</td>
<td>0.502</td>
<td>0.836</td>
</tr>
<tr>
<td>2019-08-02 03:00</td>
<td>0.470</td>
<td>0.501</td>
<td>0.642</td>
<td>0.415</td>
<td>0.512</td>
<td>2.971</td>
<td>0.502</td>
<td>0.919</td>
<td></td>
</tr>
<tr>
<td>2019-08-02 04:00</td>
<td>0.542</td>
<td>0.948</td>
<td>2.885</td>
<td>2.894</td>
<td>1.317</td>
<td>1.758</td>
<td>4.967</td>
<td>0.739</td>
<td>0.866</td>
</tr>
<tr>
<td>2019-08-02 05:00</td>
<td>0.587</td>
<td>0.659</td>
<td>0.616</td>
<td>0.824</td>
<td>0.243</td>
<td>1.352</td>
<td>4.582</td>
<td>0.460</td>
<td>0.849</td>
</tr>
<tr>
<td>2019-08-02 06:00</td>
<td>0.366</td>
<td>0.611</td>
<td>0.663</td>
<td>0.492</td>
<td>1.290</td>
<td>0.494</td>
<td>3.766</td>
<td>0.502</td>
<td>0.836</td>
</tr>
<tr>
<td>2019-08-02 07:00</td>
<td>0.470</td>
<td>0.501</td>
<td>0.642</td>
<td>0.415</td>
<td>0.512</td>
<td>2.971</td>
<td>0.502</td>
<td>0.919</td>
<td></td>
</tr>
<tr>
<td>2019-08-02 08:00</td>
<td>0.542</td>
<td>0.948</td>
<td>2.885</td>
<td>2.894</td>
<td>1.317</td>
<td>1.758</td>
<td>4.967</td>
<td>0.739</td>
<td>0.866</td>
</tr>
<tr>
<td>2019-08-02 09:00</td>
<td>0.587</td>
<td>0.659</td>
<td>0.616</td>
<td>0.824</td>
<td>0.243</td>
<td>1.352</td>
<td>4.582</td>
<td>0.460</td>
<td>0.849</td>
</tr>
<tr>
<td>2019-08-02 10:00</td>
<td>0.366</td>
<td>0.611</td>
<td>0.663</td>
<td>0.492</td>
<td>1.290</td>
<td>0.494</td>
<td>3.766</td>
<td>0.502</td>
<td>0.836</td>
</tr>
<tr>
<td>2019-08-02 11:00</td>
<td>0.470</td>
<td>0.501</td>
<td>0.642</td>
<td>0.415</td>
<td>0.512</td>
<td>2.971</td>
<td>0.502</td>
<td>0.919</td>
<td></td>
</tr>
<tr>
<td>2019-08-02 12:00</td>
<td>0.542</td>
<td>0.948</td>
<td>2.885</td>
<td>2.894</td>
<td>1.317</td>
<td>1.758</td>
<td>4.967</td>
<td>0.739</td>
<td>0.866</td>
</tr>
<tr>
<td>2019-08-02 13:00</td>
<td>0.587</td>
<td>0.659</td>
<td>0.616</td>
<td>0.824</td>
<td>0.243</td>
<td>1.352</td>
<td>4.582</td>
<td>0.460</td>
<td>0.849</td>
</tr>
<tr>
<td>2019-08-02 14:00</td>
<td>0.366</td>
<td>0.611</td>
<td>0.663</td>
<td>0.492</td>
<td>1.290</td>
<td>0.494</td>
<td>3.766</td>
<td>0.502</td>
<td>0.836</td>
</tr>
<tr>
<td>2019-08-02 15:00</td>
<td>0.470</td>
<td>0.501</td>
<td>0.642</td>
<td>0.415</td>
<td>0.512</td>
<td>2.971</td>
<td>0.502</td>
<td>0.919</td>
<td></td>
</tr>
<tr>
<td>2019-08-02 16:00</td>
<td>0.542</td>
<td>0.948</td>
<td>2.885</td>
<td>2.894</td>
<td>1.317</td>
<td>1.758</td>
<td>4.967</td>
<td>0.739</td>
<td>0.866</td>
</tr>
<tr>
<td>2019-08-02 17:00</td>
<td>0.587</td>
<td>0.659</td>
<td>0.616</td>
<td>0.824</td>
<td>0.243</td>
<td>1.352</td>
<td>4.582</td>
<td>0.460</td>
<td>0.849</td>
</tr>
<tr>
<td>2019-08-02 18:00</td>
<td>0.366</td>
<td>0.611</td>
<td>0.663</td>
<td>0.492</td>
<td>1.290</td>
<td>0.494</td>
<td>3.766</td>
<td>0.502</td>
<td>0.836</td>
</tr>
<tr>
<td>2019-08-02 19:00</td>
<td>0.470</td>
<td>0.501</td>
<td>0.642</td>
<td>0.415</td>
<td>0.512</td>
<td>2.971</td>
<td>0.502</td>
<td>0.919</td>
<td></td>
</tr>
<tr>
<td>2019-08-02 20:00</td>
<td>0.542</td>
<td>0.948</td>
<td>2.885</td>
<td>2.894</td>
<td>1.317</td>
<td>1.758</td>
<td>4.967</td>
<td>0.739</td>
<td>0.866</td>
</tr>
<tr>
<td>2019-08-02 21:00</td>
<td>0.587</td>
<td>0.659</td>
<td>0.616</td>
<td>0.824</td>
<td>0.243</td>
<td>1.352</td>
<td>4.582</td>
<td>0.460</td>
<td>0.849</td>
</tr>
<tr>
<td>2019-08-02 22:00</td>
<td>0.366</td>
<td>0.611</td>
<td>0.663</td>
<td>0.492</td>
<td>1.290</td>
<td>0.494</td>
<td>3.766</td>
<td>0.502</td>
<td>0.836</td>
</tr>
<tr>
<td>2019-08-02 23:00</td>
<td>0.470</td>
<td>0.501</td>
<td>0.642</td>
<td>0.415</td>
<td>0.512</td>
<td>2.971</td>
<td>0.502</td>
<td>0.919</td>
<td></td>
</tr>
</tbody>
</table>
What our BOTS machines looked like from CB

These look like 7-8 MB a day…

<table>
<thead>
<tr>
<th>Time</th>
<th>ABUNGSTEIN-L</th>
<th>AGRADYL</th>
<th>BSTOLL</th>
<th>BTUN-L</th>
<th>FMALTEKESKO-L</th>
<th>GHOPPY-L</th>
<th>JWORTOSKI-L</th>
<th>MALITUS-L</th>
<th>PCERF-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-08-02 00:00</td>
<td>0.542</td>
<td>0.948</td>
<td>2.885</td>
<td>2.894</td>
<td>1.317</td>
<td>1.758</td>
<td>4.967</td>
<td>0.739</td>
<td>0.866</td>
</tr>
<tr>
<td>2019-08-02 01:00</td>
<td>0.587</td>
<td>0.659</td>
<td>0.616</td>
<td>0.824</td>
<td>0.243</td>
<td>1.352</td>
<td>4.582</td>
<td>0.460</td>
<td>0.849</td>
</tr>
<tr>
<td>2019-08-02 02:00</td>
<td>0.366</td>
<td>0.611</td>
<td>0.663</td>
<td>0.492</td>
<td>1.290</td>
<td>3.766</td>
<td>0.414</td>
<td>0.938</td>
<td></td>
</tr>
<tr>
<td>2019-08-02 03:00</td>
<td>0.470</td>
<td>0.501</td>
<td>0.642</td>
<td>0.415</td>
<td>0.512</td>
<td>2.971</td>
<td>0.502</td>
<td>0.919</td>
<td></td>
</tr>
<tr>
<td>2019-08-02 04:00</td>
<td>0.542</td>
<td>0.948</td>
<td>2.885</td>
<td>2.894</td>
<td>1.317</td>
<td>1.758</td>
<td>4.967</td>
<td>0.739</td>
<td>0.866</td>
</tr>
<tr>
<td>2019-08-02 05:00</td>
<td>0.587</td>
<td>0.659</td>
<td>0.616</td>
<td>0.824</td>
<td>0.243</td>
<td>1.352</td>
<td>4.582</td>
<td>0.460</td>
<td>0.849</td>
</tr>
<tr>
<td>2019-08-02 06:00</td>
<td>0.366</td>
<td>0.611</td>
<td>0.663</td>
<td>0.492</td>
<td>1.290</td>
<td>3.766</td>
<td>0.414</td>
<td>0.938</td>
<td></td>
</tr>
<tr>
<td>2019-08-02 07:00</td>
<td>0.470</td>
<td>0.501</td>
<td>0.642</td>
<td>0.415</td>
<td>0.512</td>
<td>2.971</td>
<td>0.502</td>
<td>0.919</td>
<td></td>
</tr>
<tr>
<td>2019-08-02 08:00</td>
<td>0.542</td>
<td>0.948</td>
<td>2.885</td>
<td>2.894</td>
<td>1.317</td>
<td>1.758</td>
<td>4.967</td>
<td>0.739</td>
<td>0.866</td>
</tr>
<tr>
<td>2019-08-02 09:00</td>
<td>0.587</td>
<td>0.659</td>
<td>0.616</td>
<td>0.824</td>
<td>0.243</td>
<td>1.352</td>
<td>4.582</td>
<td>0.460</td>
<td>0.849</td>
</tr>
<tr>
<td>2019-08-02 10:00</td>
<td>0.366</td>
<td>0.611</td>
<td>0.663</td>
<td>0.492</td>
<td>1.290</td>
<td>3.766</td>
<td>0.414</td>
<td>0.938</td>
<td></td>
</tr>
<tr>
<td>2019-08-02 11:00</td>
<td>0.470</td>
<td>0.501</td>
<td>0.642</td>
<td>0.415</td>
<td>0.512</td>
<td>2.971</td>
<td>0.502</td>
<td>0.919</td>
<td></td>
</tr>
<tr>
<td>2019-08-02 12:00</td>
<td>0.542</td>
<td>0.948</td>
<td>2.885</td>
<td>2.894</td>
<td>1.317</td>
<td>1.758</td>
<td>4.967</td>
<td>0.739</td>
<td>0.866</td>
</tr>
<tr>
<td>2019-08-02 13:00</td>
<td>0.587</td>
<td>0.659</td>
<td>0.616</td>
<td>0.824</td>
<td>0.243</td>
<td>1.352</td>
<td>4.582</td>
<td>0.460</td>
<td>0.849</td>
</tr>
<tr>
<td>2019-08-02 14:00</td>
<td>0.366</td>
<td>0.611</td>
<td>0.663</td>
<td>0.492</td>
<td>1.290</td>
<td>3.766</td>
<td>0.414</td>
<td>0.938</td>
<td></td>
</tr>
<tr>
<td>2019-08-02 15:00</td>
<td>0.470</td>
<td>0.501</td>
<td>0.642</td>
<td>0.415</td>
<td>0.512</td>
<td>2.971</td>
<td>0.502</td>
<td>0.919</td>
<td></td>
</tr>
<tr>
<td>2019-08-02 16:00</td>
<td>0.542</td>
<td>0.948</td>
<td>2.885</td>
<td>2.894</td>
<td>1.317</td>
<td>1.758</td>
<td>4.967</td>
<td>0.739</td>
<td>0.866</td>
</tr>
<tr>
<td>2019-08-02 17:00</td>
<td>0.587</td>
<td>0.659</td>
<td>0.616</td>
<td>0.824</td>
<td>0.243</td>
<td>1.352</td>
<td>4.582</td>
<td>0.460</td>
<td>0.849</td>
</tr>
<tr>
<td>2019-08-02 18:00</td>
<td>0.366</td>
<td>0.611</td>
<td>0.663</td>
<td>0.492</td>
<td>1.290</td>
<td>3.766</td>
<td>0.414</td>
<td>0.938</td>
<td></td>
</tr>
<tr>
<td>2019-08-02 19:00</td>
<td>0.470</td>
<td>0.501</td>
<td>0.642</td>
<td>0.415</td>
<td>0.512</td>
<td>2.971</td>
<td>0.502</td>
<td>0.919</td>
<td></td>
</tr>
<tr>
<td>2019-08-02 20:00</td>
<td>0.542</td>
<td>0.948</td>
<td>2.885</td>
<td>2.894</td>
<td>1.317</td>
<td>1.758</td>
<td>4.967</td>
<td>0.739</td>
<td>0.866</td>
</tr>
<tr>
<td>2019-08-02 21:00</td>
<td>0.587</td>
<td>0.659</td>
<td>0.616</td>
<td>0.824</td>
<td>0.243</td>
<td>1.352</td>
<td>4.582</td>
<td>0.460</td>
<td>0.849</td>
</tr>
<tr>
<td>2019-08-02 22:00</td>
<td>0.366</td>
<td>0.611</td>
<td>0.663</td>
<td>0.492</td>
<td>1.290</td>
<td>3.766</td>
<td>0.414</td>
<td>0.938</td>
<td></td>
</tr>
<tr>
<td>2019-08-02 23:00</td>
<td>0.470</td>
<td>0.501</td>
<td>0.642</td>
<td>0.415</td>
<td>0.512</td>
<td>2.971</td>
<td>0.502</td>
<td>0.919</td>
<td></td>
</tr>
</tbody>
</table>
JWORTOSKI had a broken IPv6 config?

8/2/19 11:59:49.000 PM

{ [-]
cb_server: cbserver
computer_name: JWORTOSKI-L
direction: outbound
domain:
event_type: netconn
link_process: https://34.220.185.163/#analyze/00000003-0000-086c-01d5-4727be08fe6c/0
link_sensor: https://34.220.185.163/#/host/3
local_ip: fe80::d903:176e:3226:9023
local_port: 56999
md5: 0861726716C9610CE5F6BCF3F4858DA1
pid: 2156
process_guid: 00000003-0000-086c-01d5-4727be08fe6c
process_path: c:\windows\system32\svchost.exe
protocol: 17
proxy: false
remote_ip: fe80::21c:42ff:fe00:18
remote_port: 53
sensor_id: 3
sha256: 29f08d5f4b8d798038cb9647178a8b9c68e16dc50da850937f6e993fc7967b75
timestamp: 1564790389
type: ingress.event.netconn
}

Show as raw text
JWORTOSKI was different.
Other Endpoints...

<table>
<thead>
<tr>
<th>computer_name</th>
<th></th>
<th>MULTICAST</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ABUNGSTEIN-L</td>
<td></td>
<td>234</td>
<td>420</td>
</tr>
<tr>
<td>AGRADY-L</td>
<td></td>
<td>264</td>
<td>270</td>
</tr>
<tr>
<td>BSTOLL-L</td>
<td></td>
<td>528</td>
<td>516</td>
</tr>
<tr>
<td>BTUN-L</td>
<td></td>
<td>432</td>
<td>18</td>
</tr>
<tr>
<td>FMALTEKESKO-L</td>
<td></td>
<td>84</td>
<td>0</td>
</tr>
<tr>
<td>GHOPPY-L</td>
<td></td>
<td>294</td>
<td>0</td>
</tr>
<tr>
<td>JMORTOSKI-L</td>
<td></td>
<td>2318</td>
<td>58278</td>
</tr>
<tr>
<td>MVALITUS-L</td>
<td></td>
<td>174</td>
<td>258</td>
</tr>
<tr>
<td>PCERF-L</td>
<td></td>
<td>204</td>
<td>0</td>
</tr>
</tbody>
</table>

- Evidently CB’s “netconn” collects IPv6 by default
- Could filter this in a number of places – cb forwarder config or UF on forwarder box with indexed extractions, or indexers
- Review your data and look for anomalies like this to filter out!

**BOTS 5: ONLY IPv6! You heard it here first.**
633GB from ~4,500 Production Windows Servers
(\textsim140\text{MB} \text{a day per Server})

868GB from ~18,000 Endpoints (mostly Windows)
(\textsim48\text{MB} \text{a day per Endpoint})

NO FILTERING.
Three ways to get data into your own Splunk instance:

- Falcon SIEM Connector (detections and audit events)
- Falcon Streaming API (detections and audit events)
- Falcon Replicator (granular sensor data) usually via SQS

<table>
<thead>
<tr>
<th>FALCON STREAMING API</th>
<th>FALCON DATA REPLICA API</th>
<th>FALCON QUERY API</th>
<th>FALCON INTEL API</th>
<th>FALCON THREAT GRAPH API</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Detections</td>
<td>• Raw event data</td>
<td>• Search for IOCs, devices and detections</td>
<td>• Actors</td>
<td>• Detections</td>
</tr>
<tr>
<td>• Audit events</td>
<td></td>
<td>• Manage detections and custom IOC watch list</td>
<td>• Indicators</td>
<td>• IOC search</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• News</td>
<td>• Process metadata</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Tailored intel</td>
<td></td>
</tr>
</tbody>
</table>

FALCON API
• Three ways to get data into your own Splunk instance:
  • Falcon SIEM Connector (detections and audit events)
  • Falcon Streaming API (detections and audit events)
  • Falcon Replicator (granular sensor data) usually via SQS

Data Replicator provides hunting data.

Allows you to retain far more data in Splunk, historically (more than 7 days).

(Data newer than 7 days is available in CS console, which is OEM Splunk...)

Real time....for current customers!
Data from Crowdstrike’s Falcon Replicator…

Process (over 80%), DNS, File, etc.
Powershell Encoded

Scheduled Task Registered

Suspicious Registry Changes

Executables Running from Recycle Bin

Reconnaissance Tools

Hunting Suspicious Processes

Hunting Phishing Attacks & Malicious Attachments

Files Written to Removable Media

Rare DNS

Remote Access Tool Usage
What would endpoint collection nirvana look like?

Well, how many hours a day do your employees work?
Except for ... millennials?
~1MB per hour a “nirvana” goal.

But realistically, max 2MB per work-hour.
“Sure, but know that it’s gonna increase our Splunk ingest/storage/compute cost.”
What can we do besides audit config and filter?

Pre-Index, or “Stream” Processing!
MAGIC! From Splunk!
MAGIC!
From a partner!

- Splunk
- Kafka
- Snowflake
- Syslog
- Azure: Blob, Hub
- AWS: S3, Kinesis
- Honeycomb
- Elastic
- NFS/File
- TCP JSON
What magic?

(x10)

“Reduce by Half.”
“But I can just continue to play with audit configs at the source, and white/blacklists…”

Key Takeaway: Stream Processing centralizes and eases the config and puts YOU in complete control of your events, and where they end up.

Let forwarders forward and indexers index and search.
At-scale Windows event filtering and routing in DSP!
Cribl filtering of unwanted Crowdstrike k/v pairs!

7TB became 3TB.
(They also dropped certain classes of events...)

```json
{
  "filter": "true",
  "id": "serde",
  "description": "Filter out unwanted kv pairs",
  "conf": {
    "mode": "reserialize",
    "type": "json",
    "srcField": "_raw",
    "remove": [
      "cid",
      "name",
      "TokenType",
      "IntegrityLevel",
      "ImageSubsystem",
      "Entitlements",
      "EffectiveTransmissionClass",
      "ConfigStateHash"
    ],
    "fieldFilterExpr": ""
  }
}
```
Filtering of Common DNS Destinations!

(Variation: Alexa Top 1000)

Remember our pesky 4104 filtering issue?

MD5 Hashing of Powershell Script Block Logging Content!
Capture everything in the Message prior to "ScriptBlock"...
...and if it's the same hash, suppress it unless 10m (configurable) has elapsed.
DSP has a very rich library of functions...including hashing.

<table>
<thead>
<tr>
<th>Cryptographic scalar functions</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>md5</td>
<td>Computes and returns the MD5 hash of a byte value X.</td>
<td></td>
</tr>
<tr>
<td>sha1</td>
<td>Computes and returns the secure hash of a byte value X based on the FIPS compliant SHA-1 hash function.</td>
<td></td>
</tr>
<tr>
<td>sha256</td>
<td>Computes and returns the secure hash of a byte value X based on the FIPS compliant SHA-256 hash function.</td>
<td></td>
</tr>
<tr>
<td>sha512</td>
<td>Computes and returns the secure hash of a byte value X based on the FIPS compliant SHA-512 hash function.</td>
<td></td>
</tr>
</tbody>
</table>
Does it scale?

DSP: 5 nodes
27TB a day.

*
The Universal Forwarder: Pros and Cons

- No per-node license
- Fully supported by Splunk
- Lots of success and community help
- Efficient and secure transfer of data
- Efficient distribution of data (if architected properly)
- Less complexity
- Lots of capability besides “just logs”
Slides from .confs of yore...

The UF: It’s More Than You Think

- Process/Apps/FIM
- Perfmon
- Registry
- Wire Data
- Scripts
- Sysmon
- Logs

*Including PowerShell!
The Universal Forwarder: Pros and Cons

- No per-node license
- Fully supported by Splunk
- Lots of success and community help
- Efficient and secure transfer of data
- Efficient distribution of data (if architectured properly)
- Less complexity
- Lots of capability besides “just logs”

• It’s an agent.
People HATE agents.
The Universal Forwarder: Pros and Cons

- No per-node license
- Fully supported by Splunk
- Lots of success and community help
- Efficient and secure transfer of data
- Efficient distribution of data (if architected properly)
- Less complexity
- Lots of capability besides “just logs”

- It’s an agent
- You have to install and maintain it
- It doesn’t run on all OS’s you may have
- It only sends to Splunk*
- Improperly configured it can impact performance
- It can be used for good…or evil…
You could use Windows Event Forwarding!
You could use Windows Event Forwarding!

Events occur on endpoints, and are logged to the Windows event log.

AD SITE - US-WEST

Windows Clients register with WEF Server. Events matching a subscription are sent.

COLLECTION options

- S2S
- UF or HF

INDEXING

syslog-ng

Authentication Subscription
EventID • 4624, 4625, 4634, 4672...

Process Execution Subscription
EventID • 4688

Registry Modification Subscription
EventID • 4657, 1904, 1905, 1906...

Subscriptions are stored on the WEF Server. They control which EventIDs are sent by clients.

Maybe winlogbeat or Azure Event Hub, too…
WEF Pros and Cons

- No agent!
- No additional license cost
- Supported by Microsoft
- Can support most modern versions of Windows
- Might be the “only” option due to agentless
- Easy to configure on the endpoint via GPO
- No need to filter UF “junk” from 4688/Powershell/Sysmon
- Now supported by the Windows TA so…”officially supported” by Splunk (XML needed)

- You shift processing to a much smaller number of nodes! Latency abounds.
- You have to create and maintain a complex collection infrastructure.
- Higher network utilization due to XML:SOAP wrappers
- DCOM and RPC=++ attack surface
- Difficult to collect off campus
- No failover, no load balancing, might lose events.
- Data sources limited to “events that can log to .evtx format” so no IIS, DHCP, Windows Update, scripted collection…
- If you don’t use UF/HF then custom props/transforms
- Must use XML render
- Troubleshooting notoriously hard!
Hardening Windows Remote Management (WinRM)

Tactic: Lateral dispersion between systems via Windows Remote Management (WinRM) and PowerShell remoting

Manual operators may leverage Windows Remote Management (WinRM) to propagate ransomware throughout an environment. WinRM is enabled by default on all Windows Server operating systems (since Windows Server 2012 and above), but disabled on all client operating systems (Windows 7 and Windows 10) and older server platforms (Windows Server 2008 R2).

PowerShell Remoting (PS Remoting) is a native Windows remote command execution feature that’s built on top of the WinRM protocol.

If WinRM has ever been enabled on a client (non-server) operating system, then the following configurations will exist on an endpoint, and will not be remediated solely through the PowerShell command noted in Figure 20.

- WinRM listener configured
- Windows Firewall exception configured

These items will need to be disabled manually through the commands in Figure 23 and Figure 24.

PowerShell:

```
Disable-PSRemoting -Force
```

Note: Disabling PowerShell Remoting does not prevent local users from creating PowerShell sessions on the local computer or for sessions destined for remote computers.

After running the command, the message recorded in Figure 21 will be displayed.

---

**Figure 20.**

PowerShell Command to disable WinRM / PowerShell Remoting on an endpoint.

---

**Figure 21.** Warning message after disabling PSRemoting.

```
PS C:\WINDOWS\system32> Disable-PSRemoting -Force
WARNING: Disabling the session configurations does not undo all the changes made by Enable-PSSessionConfiguration cmdlet. You might have to manually undo the changes in the following:
1. Stop and disable the WinRM service.
2. Delete the listener that accepts requests on any IP address.
3. Disable the firewall exceptions for WS-Management communications.
4. Restore the value of the LocalAccountTokenFilterPolicy to 0, which restricts remote access to members of the Administrators group on the computer.
```

---

**Figure 22.** Command to set WinRM configuration.
WinEventLog://ForwardedEvents

blacklist1 = EventCode="566" Message="Object Type:s+(?!groupPolicyContainer)"
blacklist2 = 4656,4658,4660-4663,4665-4667,4673,4690,4793,4907,4932,4933,4985
blacklist3 = 5061,5058,5145,5152,5154,5156-5158
blacklist4 = 26401,36886
blacklist5 = EventCode="4688" Message="(?i:New Process
Name:).+(?:SplunkUniversalForwarder\bin\splunk.exe).+(?:SplunkUniversalForwarder\bin\btool.exe).+(?:Agent\bin\MonitoringHost.exe)"
blacklist6 = 2002,4614,4664,4675,4700-4702,4717,4779,4905,4931,4933,4944,4945,4957,5012,5024,5056,5058,5059,5061,5379,5440,5442,5444,5447,5448,5450,5457,5478,5632,5633,588,5890,6278,6419,6421,6422,7001,7036,7043
blacklist7 = EventCode="4674"
Message=".*\s*Account\sName:.*specadmin.*Process\sName:.*\W\Ss*SysWOW64\W\WmiPrvSE.exe|.*\Ss*System32\W\WmiPrvSE.exe"
current_only = 0
disabled = 0
evt_dc_name =
evt_dns_name =
evt_resolve_ad_obj = 0
host = WinEventLogForwardHost
renderXML=false
interval = 60
sourcetype = WinEventLog:ForwardedEvents
start_from = oldest
suppress_sourcename=true
suppress_keywords=true
suppress_type=true
suppress_task=true
suppress_opcode=true
suppress_text=true

Prevents event latency, but actually isn’t ideal, and isn’t CIM compliant…
We don’t have a lot of examples of successful WEF/WEC deployment at scale. 😞
what about…
You could use cloud storage*

- Microsoft WEF
- WEC
- Azure Event Hub
- Splunk

• Windows Logs

@Andrew Huddleston

WEF support is currently in preview and still has some limitations. Contact me directly if you would like to join, and we can discuss whether the current support would work for you.

As an alternative, you can continue to use CEF and winlogbeat and connect it to Sentinel using Logstash and the Logstash Log Analytics output plugin.

~ Ofer

Best Response confirmed by Andrew Huddleston (Frequent Contributor)
You could use cloud functionality!

* You could use cloud functionality!
  
  osquery ➔ AWS ➔ Kinesis Streams or Firehose*

- Splunk Add On for Amazon Kinesis Firehose
- Splunk Input for Kinesis Streams
- SQS-based S3 input

---

As of version 1.7.4, osquery can log results directly to Amazon AWS Kinesis Streams and Kinesis Firehose. For users of these services, osqueryd can eliminate the need for a separate log forwarding daemon running in your deployments.

**Configuration**

The Kinesis Streams and Kinesis Firehose logger plugins are named `aws_kinesis` and `aws_firehose` respectively. They can be enabled as with other logger plugins using the config flag `logger_plugin`.

Some configuration is shared between the two plugins:

```
--aws_access_key_id VALUE
--aws_profile_name VALUE
--aws_region VALUE
--aws_secret_access_key VALUE
--aws sts arn_value VALUE
--aws sts region VALUE
--aws sts session_name VALUE
--aws sts timeout VALUE
--aws enable proxy VALUE
--aws proxy scheme VALUE
--aws proxy host VALUE
--aws proxy port VALUE
--aws proxy username VALUE
--aws proxy password VALUE
```

When working with AWS, osquery will look for credentials and region configuration in the following order:

1. `~/.aws/config` or `aws config`
2. `~/.aws/cfg` or `aws config`
3. `~/.aws/credentials` or `aws creds`
4. `~/.bashrc` or `aws creds`
5. `~/.bash_profile` or `aws creds`
6. `~/.bashrc` or `aws creds`
7. `~/.bash_profile` or `aws creds`
You could pay for and use Microsoft Defender ATP!

- ATP capability built into Windows 10, later server versions. Installable on 7, 8, 2016, 2012
- Needs E5 license for desktops and Azure Security Center licenses for servers
- MacOS (but signature based)
- No CIM mapping

Alerts
- Detections
- Raw “Hunting” Events
- (not Win Events)

DSP or ATP Modular Input (Alerts and Detections)
Hands On! **Encoded Powershell Logs!**
How did Violent Memmes avoid C2 detection during execution?

Sourcetypes: Microsoft Sysmon and/or WinEventLog:Security

MITRE ATT&CK: Execution
T1086: Powershell
T1043: Commonly Used Port
T1132: Data Encoding
T1172: Domain Fronting
The adversary used domain fronting to obfuscate the origin of their command and control (C2) traffic. Clues exist that provide insights into the HTTP host header used to mask the true origin of the traffic. What is the host header that is used by the adversary?

(7 correct!)
(Hands On Redacted)
What’s New?
What’s new with Sysmon?

- DNS Logging with EventCode 22
- Our TA for Sysmon is Endpoint CIM compliant
- The Github version supports Sysmon 10.x
- Researchers publishing new rulesets for granular detections:
  - UAC Bypass
  - Chinese/Vietnamese/Iranian keyboard layout connecting to server
Updated Olaf/TaySwift Sysmon to Eliminate this:

https://
New “SEDCMD” Cleanups in Win TA 6.0!

[source::XmlWinEventLog:Security]

##### Explanation for SEDCMD Extractions #####

- **windows_security_event_formater**: This will replace all values like "Account Name: -" to "Account Name:"
- **windows_security_event_formater_null_sid_id**: This will replace all values like "Security ID:NULL SID" to "Security ID:" and all values like "Logon ID:0x0" to "Logon ID:"
- **cleansrcip**: This will replace all values like "Source Network Address: ::1" or "Source Network Address:127.0.0.1" to "Source Network Address:"
- **cleansrcport**: This will replace all values like "Source Port:0" to "Source Port:"
- **remove_ffff**: This will replace all values like "Client Address: ::ffff:10.x.x.x" to "Client Address:10.x.x.x" which Addresses most of theIpv6 log issues
- **clean_info_text_from_winsecurity_events_certificate_information**: This will delete all the information text at the end of event starting from "Certificate information is..." before indexing
- **clean_info_text_from_winsecurity_events_token_elevation_type**: This will delete all the information text at the end of event starting from "Token Elevation Type indicates..." before indexing
- **clean_info_text_from_winsecurity_events_this_event**: This will delete all the information text at the end of event starting from "This event is generated..." before indexing
- **cleanxmlsrcport**: This will replace all values like <Data Name='IpPort'>0</Data> to <Data Name='IpPort'></Data> in XmlWinEventLog:Security
- **cleanxmlsrcip**: This will replace all values like <Data Name='IpAddress'>::1</Data> or <Data Name='IpAddress'>127.0.0.1</Data> to <Data Name='IpAddress'></Data> in XmlWinEventLog:Security

##### SEDCMD Extractions #####

```python
#SEDCMD-windows_security_event_formater = s/(?m)(^s*[^:]+:)*s+-?$/g
#SEDCMD-windows_security_event_formater_null_sid_id = s/(?m)(s*NULL SID$s+0x0)/g
#SEDCMD-cleansrcip = s/(Source Network Address: (::1|127\.:0\.:0\.:1))/g
#SEDCMD-cleansrcport = s/(Source Port:)*$/g
#SEDCMD-remove_ffff = s/:ffff:/g
#SEDCMD-clean_info_text_from_winsecurity_events_certificate_information = s/Certificate information is only\[Sis\n\]g
#SEDCMD-clean_info_text_from_winsecurity_events_token_elevation_type = s/Token Elevation Type indicates\[Sis\n\]g
#SEDCMD-clean_info_text_from_winsecurity_events_this_event = s/This event is generated\[Sis\n\]g
```

## For XmlWinEventLog:Security

```python
#SEDCMD-cleanxmlsrcport = s/<Data Name='IpPort'>0</Data> to <Data Name='IpPort'></Data>
#SEDCMD-cleanxmlsrcip = s/<Data Name='IpAddress'>::1</Data> or <Data Name='IpAddress'>127.0.0.1</Data> to <Data Name='IpAddress'></Data>
```

Non-destructive truncate of Message block
cmdReporter is an endpoint detection and response tool for macOS.

Using native built-in resources, it collects the data IT security teams need to hunt threats on macOS computers in real time.

8MB daily on average, 14MB if highly granular network connections enabled (If a process changes prefs, elevates privs, or makes network connections info is sent)

25,000 mac endpoints so far...

Thanks Dan Griggs!
What cmdReporter does

cmdReporter

Event filters

Translation to JSON

~48,800 MB/day

macOS security info

~8 MB/day

splunk>
macOS 10.15b1 security data in Splunk cross-platform dashboard

Authentications and Changes

Authentications by Action

Authentications by User

Authentications by App

Asset Authentications

<table>
<thead>
<tr>
<th>_time</th>
<th>src</th>
<th>dest</th>
<th>action</th>
<th>app</th>
<th>count</th>
<th>user</th>
<th>src_user</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-06-11</td>
<td>Dan_macbook_pro</td>
<td></td>
<td>success</td>
<td>sudo</td>
<td>370</td>
<td>root</td>
<td>dan</td>
</tr>
<tr>
<td>2019-06-16</td>
<td>Dan_macbook_pro</td>
<td></td>
<td>success</td>
<td>sudo</td>
<td>186</td>
<td>root</td>
<td>dan</td>
</tr>
<tr>
<td>2019-06-23</td>
<td>Dan_macbook_pro</td>
<td></td>
<td>success</td>
<td>sudo</td>
<td>160</td>
<td>root</td>
<td>dan</td>
</tr>
</tbody>
</table>

(infosec app)
What evidence can we find surrounding previous infiltration from Violent Memmes?

Sourcetypes: Microsoft Sysmon (or any other source that provides DNS query info), Carbon Black Response

MITRE ATT&CK: Establish and Maintain Infrastructure, Execution

T1333 Dynamic DNS (pre ATT&CK)
T1085 Rundll32
There is evidence in the logs that the Violent Memmes have been on the Frothly network before. If you follow that evidence, what is the Base64 string of the fully qualified domain name (FQDN) the adversary communicates with?

(ZERO correct! 80 wrong attempts.)
(Hands On Redacted)
Endpoints remain one of the most important security data sources.

There are many rich and varied endpoint sources both free and commercial you can ingest, and they are critical for advanced detection.

Not everything is critical to collect and we now have tools to help you decide what is best for you!
Thank You!

Go to the .conf19 mobile app to
RATE THIS SESSION