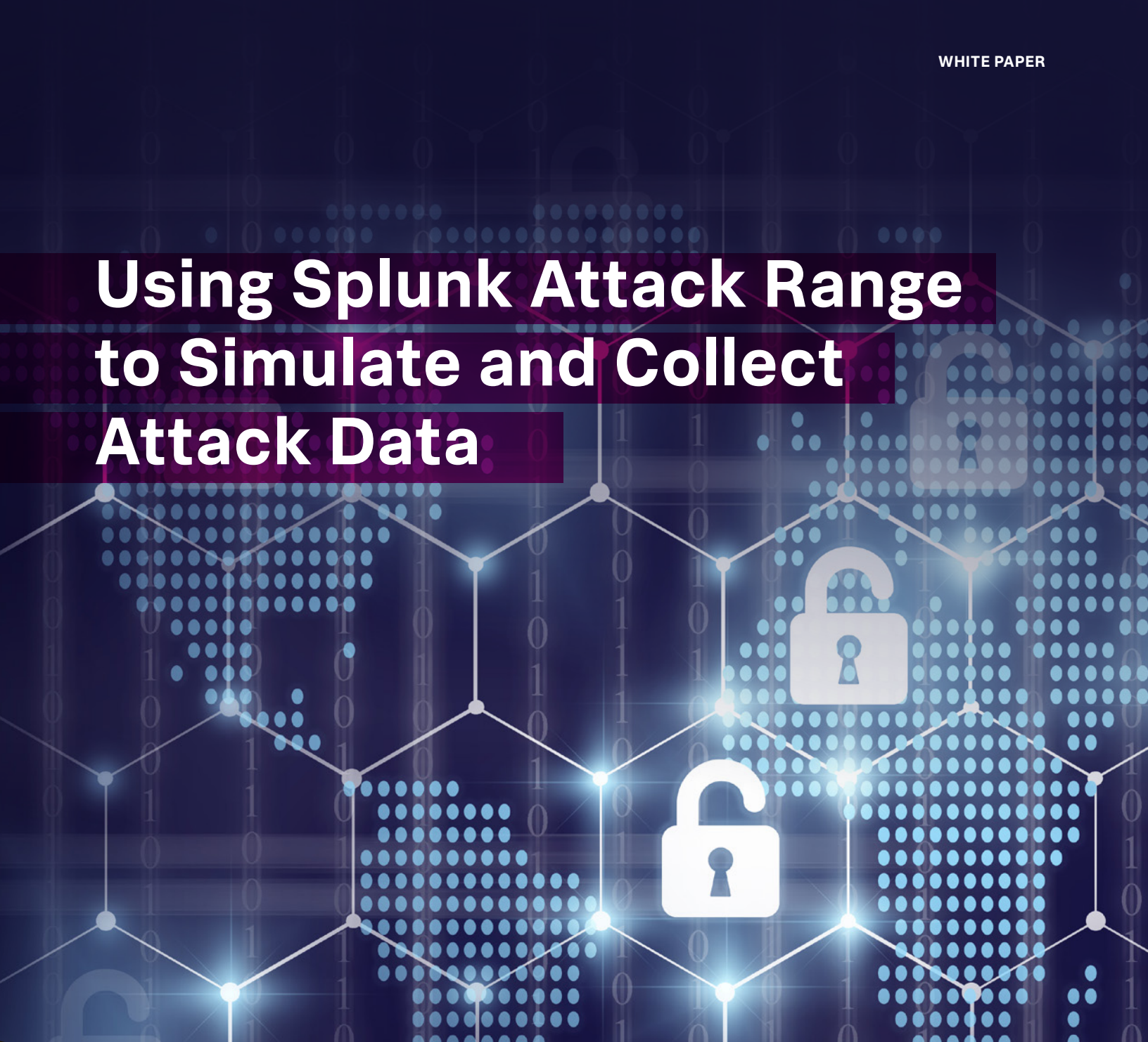
WHITE PAPER

Using Splunk Attack Range to Simulate and Collect Attack Data



Every defender's challenge includes keeping up with the evolving threat landscape

splunk> turn data into doing™

Every defender's challenge includes keeping up with the evolving threat landscape. This becomes even more difficult as enterprises adopt multiple disparate technologies that require a number of resources with specialized training, making it difficult to share, coordinate and implement new defenses. These new defenses manifest as a result of implementing artifacts that stop or mitigate threats — usually in the form of signatures, detections, playbooks or other measures.

Although some organizations do have diverse teams that can work in coordination to successfully address threats as they happen, this is not the case for most companies.

While internet resources may help, understanding and replicating public exploits and binaries requires skills that are not a part of most analysts' repertoires. Unfortunately, the security industry isn't much help. Companies tend to protect their defense artifacts (such as antivirus or firewall signatures), as they are often a source of revenue. Even in groups or organizations that share such information, it comes in very specific forms that do not help analysts see the whole picture (i.e., a binary that may contain malicious code instead of the actual exploit, a signature of attack instead, or, occasionally, a PCAP of its traffic). Further, in most enterprises, the simulation or replication of attack code is simply banned.

The bottom line is that analysts must constantly cobble together pieces from disparate, scarce, obfuscated sources of information in their quests to make sense of threats and create defense artifacts.

The Splunk Attack Range and Simulation Engine

The Splunk Security Research Team has developed a framework that allows replication, verification, and data production of attacks in a shareable, community-friendly fashion. The Splunk Attack Range allows the use of adversarial simulation engines — along with tools for measurement, translation and recording in defense technologies that help streamline the process of creating defense artifacts (signatures, detection, investigation, analytics, playbooks, and so on). It also provides simulation and verification of the entire cycle of a threat.

This framework allows the security analyst to replicate and generate data as close to ground truth as possible, in a format that allows the creation of detections, investigations, knowledge objects, and defense playbooks in Splunk. This data includes things such as logs, network captures and endpoint events derived from either known attack-simulation engines or recent exploit code. This attack simulation gets recorded and piped into Splunk (using the Attack Range) for further analysis, detection and investigation, as well as to kick off response playbooks (via Splunk Phantom).

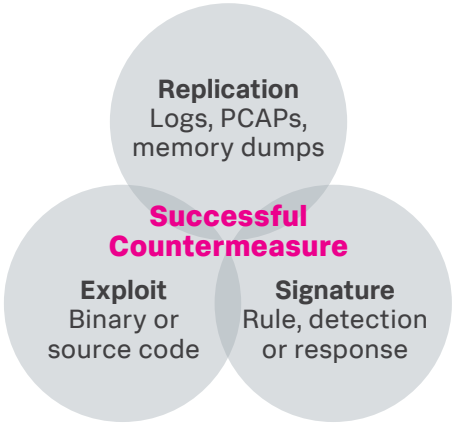
The above allows analysts to produce data and to:

- Visualize and record attacks.
- Translate attacks into measurable data.
- Drive defense artifacts based on produced data (firewall, endpoint, Snort, etc.).
- Test malicious/exploit code in a safe and isolated environment.
- Translate defense artifacts into the Splunk environment (detection, investigation, analytics, playbooks).
- Share artifacts (detection/investigation using Splunk Search Processing Language, Splunk apps and data models, both within the enterprise and within the community).

Analysts can now streamline the multiple facets of the attack cycle in a single framework, improving the speed of response and the production of defense artifacts or countermeasures. It also helps to have community-wide criteria for tactical SIEM application, wherein a common framework allows any organizations, analysts or researchers to share information and collaborate when facing past, current and future threats.

This framework also aims to simplify the logistics of the entire attack/countermeasure cycle. It follows the CI/CD (DevOps continuous integration/delivery) principles, allowing analysts to perform updates, modifications and replications based on code as infrastructure-delivery engines (Terraform, Ansible, Vagrant, etc.). This eliminates the need for on-premise investment in test equipment and expertise required for specific situations.

Attack/Countermeasure Replication Gradient

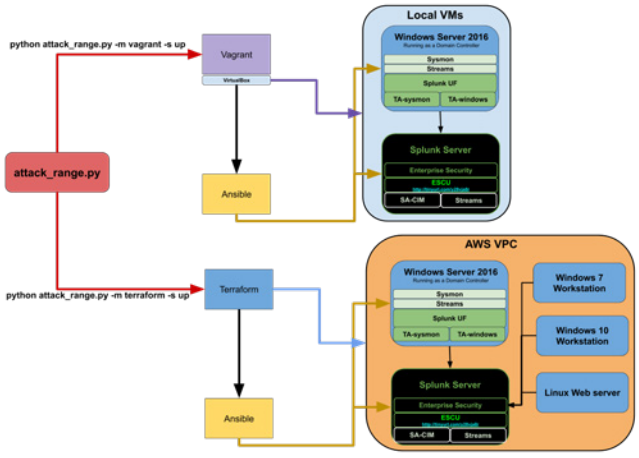


Design Considerations

The Splunk Attack Range and Threat Simulation is designed to be light and operated from a standard workstation. Most of the tools involved in the operation of the simulations will be automatically installed and configured for attack replication and recording. The tool **attack_range.py** depends on three major pieces of software (for more details see "Requirements"):

1. Vagrant/Virtualbox: This lets you run local testing using virtual machine, all hosts are executed on your local machine.
2. Terraform: In order to run cloud testing, all hosts are created and network inside an AWS VPC.
3. Ansible: This orchestrates the build process across both environments for all attack-range assets.

To get a better sense on how **attack_range.py** builds a range, please refer to the logical diagram below:



Requirements

The following are the basic requirements to get started using this framework:

Vagrant (Local range)

- Vagrant Command Line Tool installed (see guide).
- A machine with 16GB of memory and at least 100GB of free HD space
- Be able to access https://app.vagrantup.com

Terraform (Cloud range)

- Brew install Terraform CLI on OSX (see other platforms)
- Brew install AWS CLI on OSX otherwise (see guide)
- Get AWS API token. Using the AWS CLI, run **aws configure**

Other considerations:

- Supports Linux and Windows Endpoints and Servers, including:
  - Windows 10 and Windows 7 workstations
  - Windows Server 2008, 2012
  - Windows Server 2016 as a domain controller
- Is operated by a CLI tool, **attack_range.py**, built in python3
- Defines vulnerable environment as code
- An attack-simulation tool or framework is necessary to streamline the attack/exploitation process. This can be done via frameworks such as Red Canary, MITRE Caldera, or AttackIQ, or manually, since many exploits and malicious code are not always ported to these frameworks in a timely fashion when new zero-day exploits are discovered and there is no available community data. Today Splunk Research runs its own simulation tool called **attack_simulation.py** base on AttackIQ.

What's Included With attack_range:

- Builds Splunk Enterprise server with the following add-ons:
  - Splunk Stream
  - Splunk Add-on for Microsoft Windows
  - Add-on for Microsoft Sysmon
  - Splunk Common Information Model (CIM)
  - Splunk ES Content Update

- Configures Windows domain controller and installs:
  - RSAT Active Directory Admin Center
  - Active Directory Domain Services, including management tools
  - Splunk Universal Forwarder
  - Splunk Stream
  - Splunk Add-on for Microsoft Windows
  - Add-on for Microsoft Sysmon
  - Sysmon
- Configures Windows endpoints with:
  - Splunk Universal Forwarder
  - Splunk Stream app
  - Splunk Sysmon TA (TA-sysmon)
  - Splunk Windows TA (TA-windows)
  - Sysmon

Data Models

Below are some examples of Splunk data models that the range populates when an attack or log activity is generated:

- **Network_Traffic:** Splunk Stream
- **Network_Resolution:** Splunk Stream
- **Change:** Splunk Stream
- **Web:** Splunk Stream
- **Authentication:** Splunk Stream, TA-windows
- **Certificates:** Splunk Stream
- **Email:** Splunk Stream

The use and creation of data models is a parallel process of understanding and creating new detections and investigations as data is created, indexed and analyzed in an automated fashion. Any Splunk data model relevant and related to the specific threat being researched can be integrated with this framework. These data models help streamline the ingestion and processing of data.

Things to Consider When Using This Framework

Many times, even though the data is coming from a known source or application, the actual signature of the attack is hidden, obfuscated or simply not obvious. As such, the analyst may need to run additional analytics against an attack dataset to extract a signature. When defining signatures, the best way to get started is with simple Splunk queries and build them up from there.

Good practices in building searches in Splunk require event timestamps to be included (_time). Time allows the understanding and detection of events that may be proximate in time and or concurrent to the threat being researched. Creating tables with event timestamps fields and other common values should be the first step in building up detection and investigation searches. Once you've discerned the basic timeline of events, the process of labeling events based on security metadata should follow. This will help you start visualizing the attack cycle.

The Splunk Attack Range framework has been built to pair its data collection with an attack-simulation tool (**attack_simulation.py**) developed by the Splunk Security Research Team. This attack-simulation tool uses MITRE ATT&CK and Lockheed Martin Cyber Kill Chain as a TTP classification reference. These are comprehensive references of adversary tactics, techniques, and procedures (TTPs) based on field observations. These mapping frameworks provide research analysts with a referential body of knowledge that can be used in the process of labeling threat/attack data. Analysts can label data, as well as provide context for search detections, investigations, and playbook outcomes. The products of this process can then be shared in a community industry-wide accepted language.

Some of the MITRE ATT&CK items include:

- **Mitre_attack_phases:** Name of MITRE ATT&CK phases, such as Defense Evasion, Execution, and so on
- **Mitre_attack_id:** Technique ID (such as "T1117")
- **Kill_chain:** Attack link within the kill chain (Discovery, Exploitation, Installation, and so on)

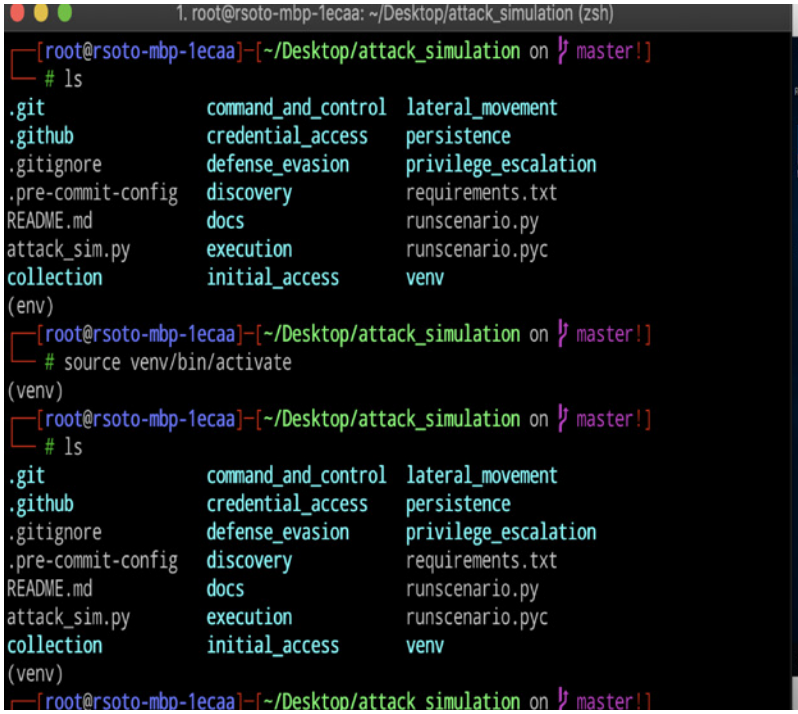
In addition to the above items, the Splunk Attack Range framework provides the following items to facilitate sharing and collaboration.

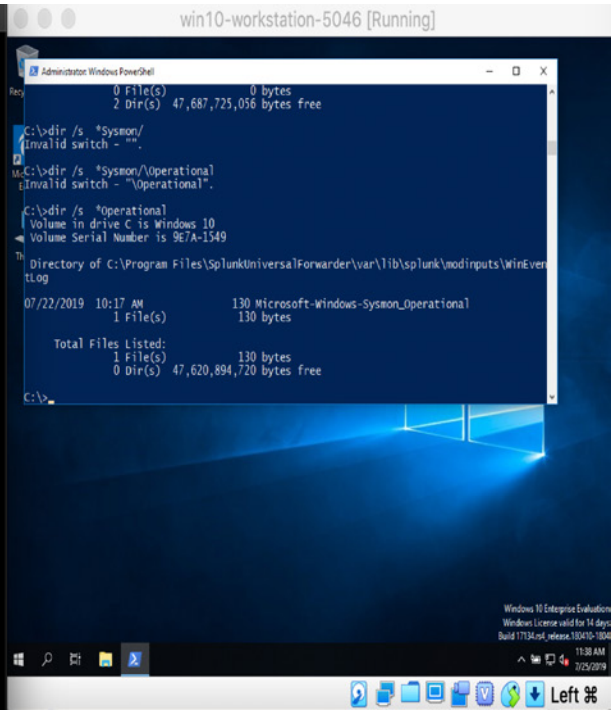
- **Technique:** Description of the attack technique
- **Attack Phase:** Attributed attack phase
- **Attack Scenario:** Real-world related data
- **Asset Information:** Specific information of targeted assets (Windows 10)
- **Response:** Recommend actions against the threat (investigation, detection, playbooks, and so on)
- **Detections:** Specific information about detection
- **External references:** Any community industry reference that can be used for addressing threat

An Example of Splunk Attack Range and Simulation Engine Framework Use In a Local Environment

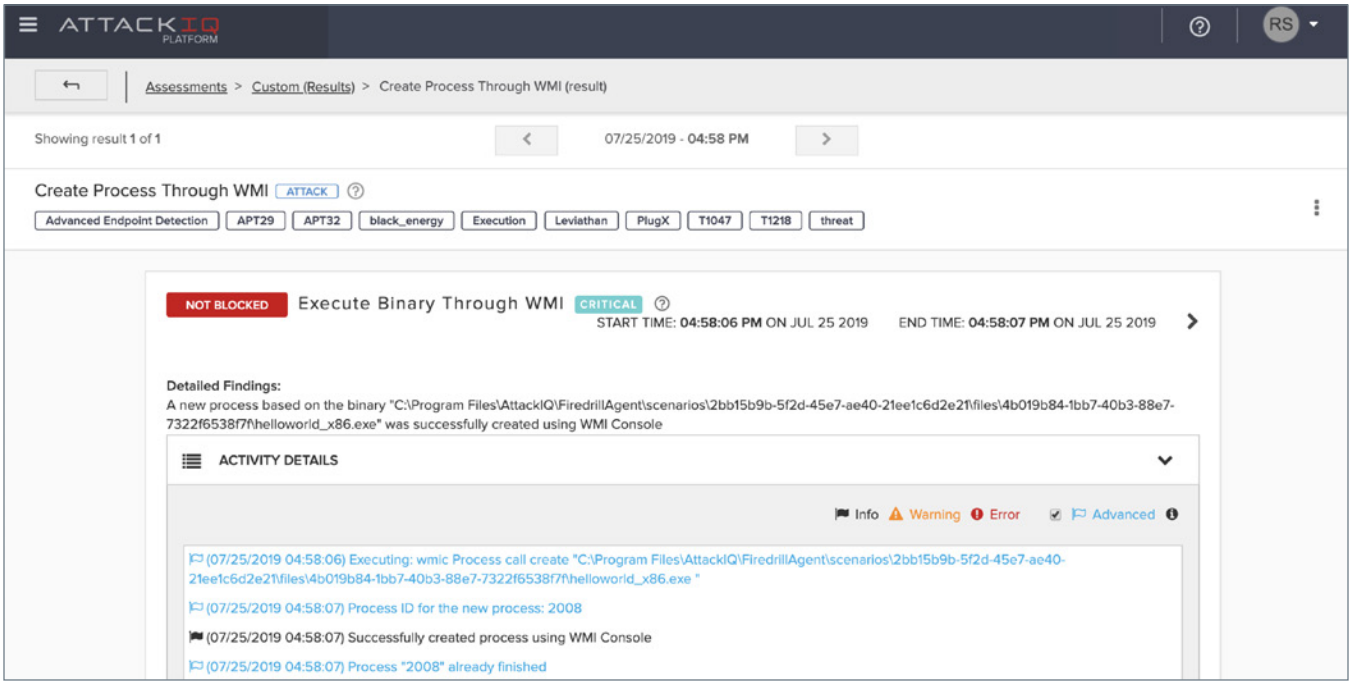
Once attack_range has been downloaded from Github and its dependencies installed, the analyst can quickly fire up the attack scenarios and proceed to replicate, index, analyze and create detection and investigatory searches.

Using Vagrant to Start a Local Windows10 Instance

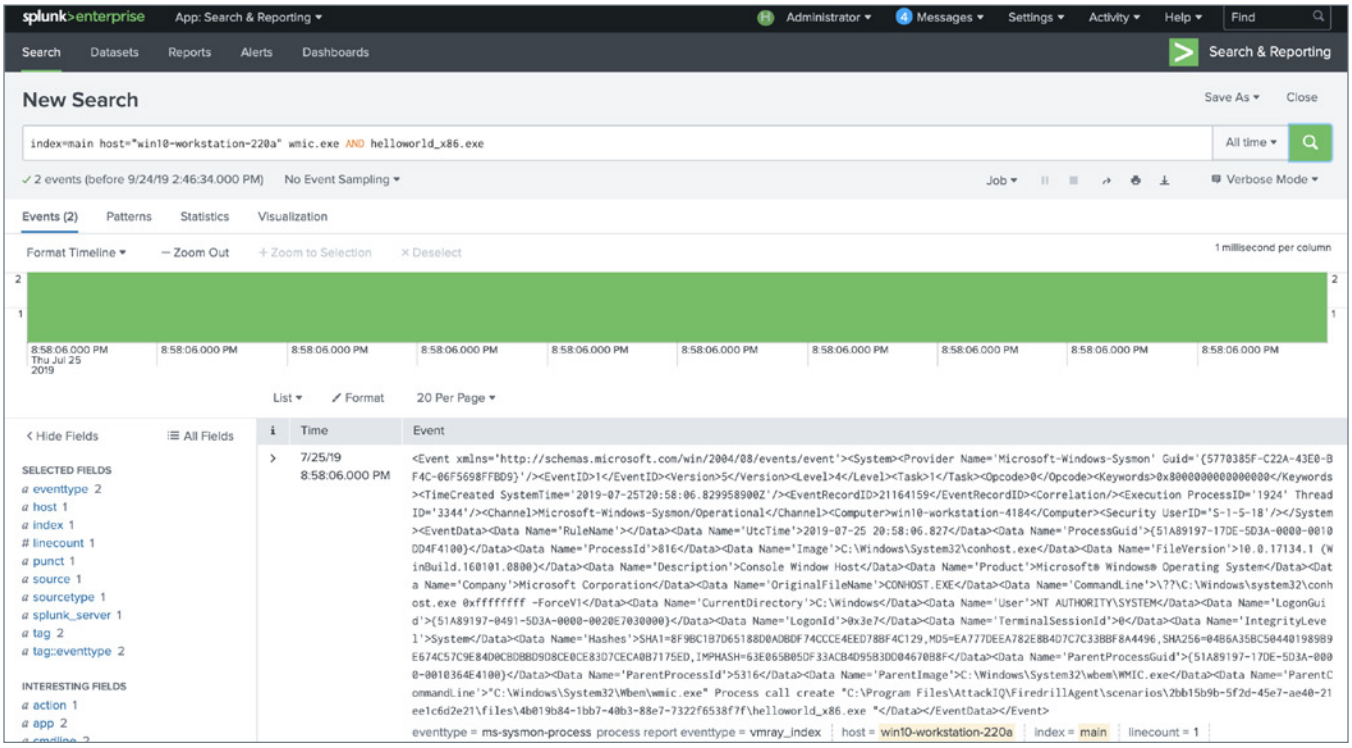




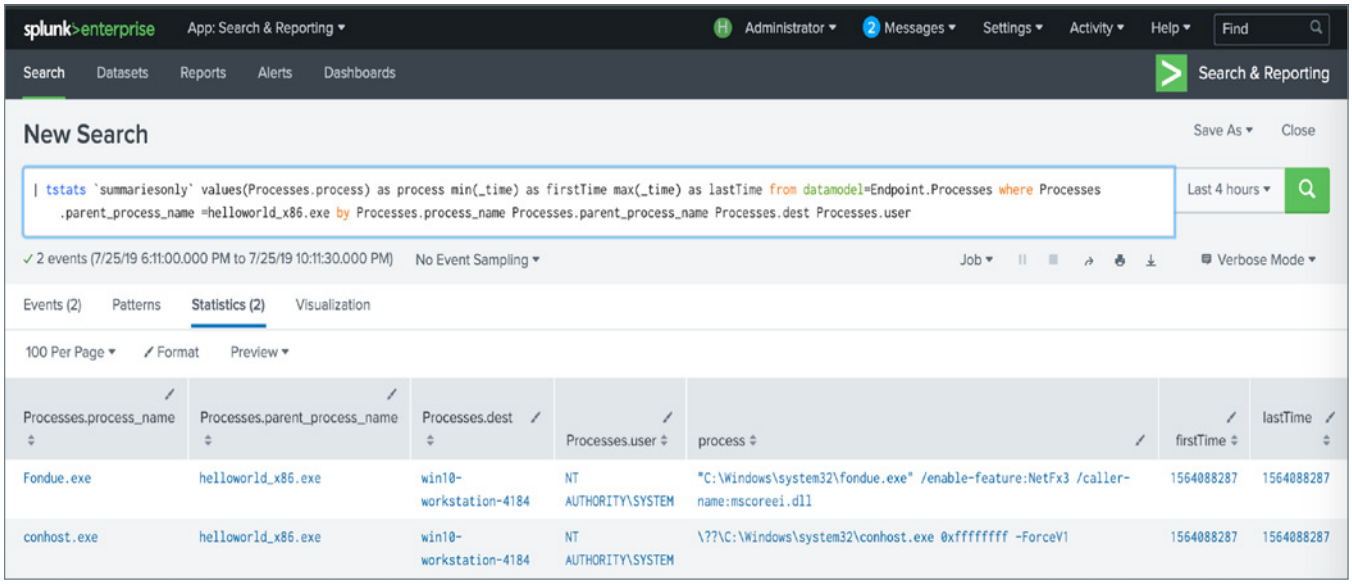
AttackIQ GUI



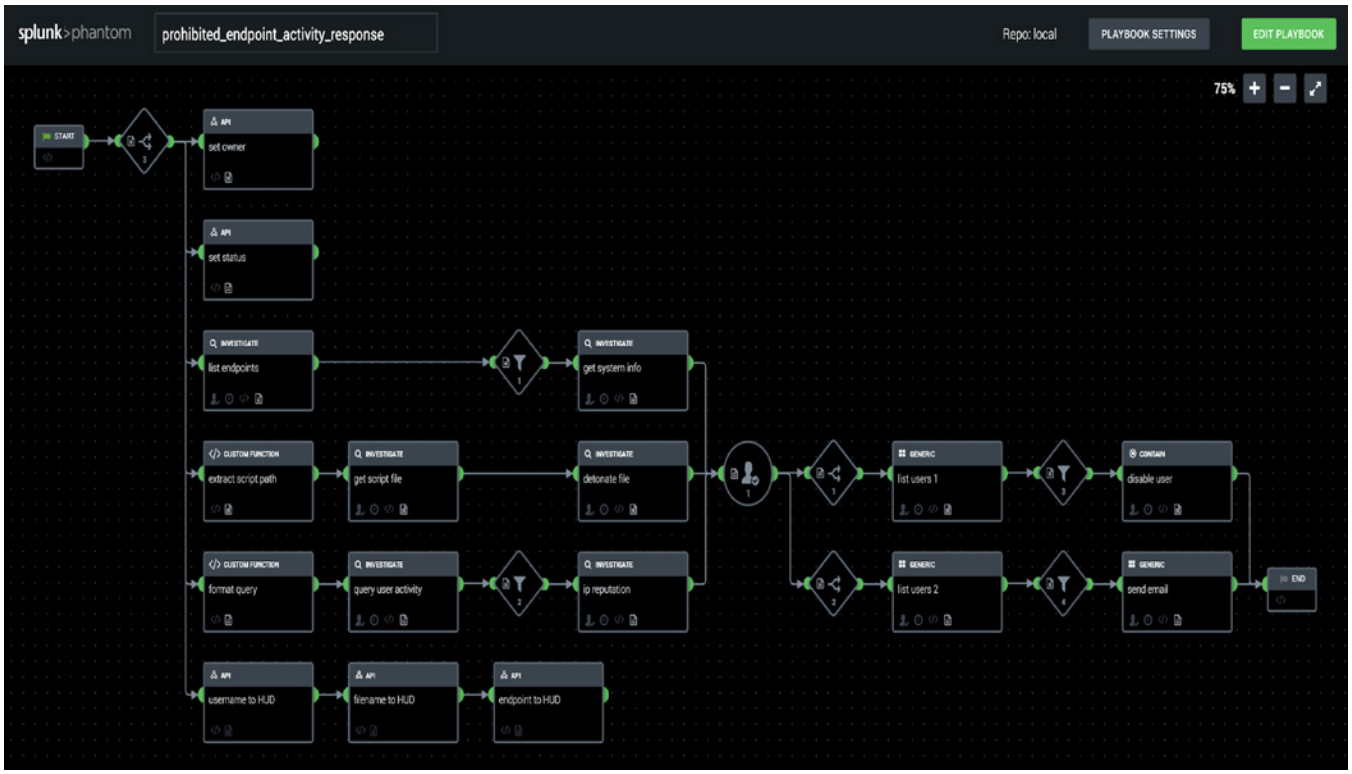
The attacks may also be executed via command line, using the **execution framework** of Atomic Red Team or other attack simulation tools. The next step is to record and index the information recorded from the attack into a Splunk instance for further analysis.



As seen above, the attack data is retrieved via a simple Splunk search that allows for the initial visualization and identification of attack data. At this point, the analyst can build detection and investigation searches, as well as other knowledge objects and countermeasures. The screenshot below shows an investigation search based on attack data and initial Splunk search queries.



Finally, countermeasures can be produced with this data and applied via Splunk Phantom. Below is an example of a Splunk Phantom playbook designed for addressing this attack technique.



A Community Effort

This framework allows security analysts and content developers to replicate and generate data as close to the truth as possible in a format that allows for the speedy creation of detections, investigations, knowledge objects and defense playbooks in Splunk. Our intent with open sourcing Splunk's Attack Range is to overcome the barriers to sharing information between analysts from multiple enterprises and organizations by creating an automated process for replication of threats, production of data and creation of attack countermeasures.

The sharing of information based on a common industry and community-accepted framework will help both the community and the industry to address threats in a more efficient manner. It can also help to cut the time and cost involved in preparing for and responding to current and future threats. You can get the latest version of the attack_range at: https://github.com/splunk/attack_range.

You can get the latest version of the attack_range at: https://github.com/splunk/attack_range.