Malware Analysis:

Suricata & Splunk for better rule writing

Anthony Tellez, CISSP, CEH, CNDA
Lead Consultant, Emerging Technologies | Splunk
November 2017
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, Splunk>, Listen to Your Data, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2017 Splunk Inc. All rights reserved.
Biography
Who Am I?

• Time In Service @ Splunk > 3 Years
• Previous: U.S. Gov Contractor, Geospatial Analyst
• Specializations
  • Cryptography
  • Information Security – Red Team
• Data Scientist
  • Security Analytics
  • Data Visualization
• Responsible for the relationship between emerging technologies and field organization
  • Acquisitions
  • Incubation
  • Product Development
• https://github.com/anthonygtellez/
Overview & Objectives

Why am I giving this talk?
Agenda
This is where the subtitle goes

- Objectives of the talk
- Building a “test range”
- Finding “Bad Data”
- Exploratory Data Analysis
- Iteration & Experimentation
- Operationalizing Insights to understand rules
Objective

Why did I commit to two talks?
Problem: Training security analysts on spotting bad activity in network traffic is difficult, in many cases they only see the out of the box alerts of many IDS/IPS tools.

This leads to analysts becoming overly reliant on the out of the box threat intelligence solutions and signatures found in various products. These analysts sometimes will learn through experience how to adjust signatures for their own environment to reduce noise or alert on certain activities.

The purpose of this talk is to provide a training framework to bootstrap an analyst so they can hunt like a pro and contribute back to the organization.

Objectives of the Talk
AKA Why did I commit to do two talks?
Building A Test Range

Do you really need your own lab?
Building A Test Range
Do you really need a lab?

Advice

- A test range doesn’t have to mirror your own Enterprise network, committing to a lab environment of this size becomes cost prohibitive and eventually requires staff to manage.
- Using pcaps you can easily sample in a virtual machine, micro instance (AWS) or raspberry pi (yes, Suricata runs on the pi!)
- MHN Honeynets/Honeypots can be troublesome to manage long term.
  - AWS micro (Free Tier) instances will last about 30ish days before all of the disk space is consumed.
  - ~30 days worth of data is usually more than enough to hunt.
- Summary: Choose the method that is going to be easiest for you to learn.
Building A Test Range
Do you really need a lab?

For purposes of this talk, we will look at the VM/Raspberry Pi option:

- Virtual Machine
  - Ubuntu 16.04
  - Suricata 4.0.1
  - Splunk 7.0.0
  - Oinkmaster 2.0
Finding Bad (Labeled) Data

Everything is infected.
Finding Bad (Labeled) Data
If everything is under attack how do you write signatures?

- One of the most difficult tasks as a newbie analyst is figuring out what behaviors are bad.
- There are obvious signs such as: attempted access to services on privileged ports.
  - How many analysts know all of the privileged ports & services that normally run on those ports?
  - What is the likelihood that anyone wants to see alerts every time SSH, RDP or TELNET connections are seen in the DMZ?
- Another key problem with this approach is the data is unlabeled for the most part without some form of threat intelligence or incident response process.
  - Labeled malware samples are freely available from various sources:
    - http://www.malware-traffic-analysis.net
    - ^ This resource provides them in PCAP format
Exploratory Data Analysis

What am I even looking at?
Create a directory called malware_zips

wget the list of urls:

- $ while read line; do wget "$\{line\}"; done < ../wget_list.txt
Create a list of all the zip files

- `ls > ../all_zips`

Decrypt the zip files with the password

- `while read line; do unzip -P <password> "${line}"; done < ../all_zips`
Exploratory Data Analysis
Getting the data to the VM

- Create a list of all the zip files
  - `ls > ../all_zips`

- Decrypt the zip files with the password
  - `while read line; do unzip -P <password> "${line}"; done < ../all_zips`

Get the password from: [http://www.malware-traffic-analysis.net/about.html](http://www.malware-traffic-analysis.net/about.html)
Create a list of all the pcap files
• find ./ -name "*.pcap" >> ../all_pcaps

Start Suricata in socket mode
• /usr/bin/suricata -c /etc/suricata/suricata.yaml \ --pidfile /var/run/suricata.pid -v --unix-socket=/var/run/666.socket -D

Exploratory Data Analysis
Getting the data to the VM
Exploratory Data Analysis

Getting the data to the VM

Run a Test:

- `/usr/bin/suricatasc /var/run/666.socket
  -c "pcap-file /root/malware_zips/2017-03-08-Hancitor-malspam-traffic.pcap /tmp"

- `{"message": "Successfully added file to list", "return": "OK"}

Note: Fully qualified path is required otherwise you will see the following error:

- `{"message": "File does not exist", "return": "NOK"}`
Prepare to read malware Samples:
• mkdir /tmp/suricata_malware

Modify PCAP list to have fully qualified path:
• Using Vi substitution
  %s/.\//\//home\atellez\malware_zips\//g
Send the list of PCAPs to the socket for reading and output eve.json to a different directory:

while read line; do /usr/bin/suricata_sc /var/run/666.socket -c "pcap-file ${line} /tmp/suricata_malware"; done < ../all_pcaps
Exploratory Data Analysis
Getting the data to the VM

- Check the output once complete
  - /root/suricata_malware/eve.json
  - tail /root/suricata_malware/eve.json

```
root@malware-analysis:~/malware_zips# ls -lh /root/suricata_malware/
total 214M
-rw-r----- 1 root root 210M Mar 10 06:22 eve.json
-rw-r----- 1 root root 2.1M Mar 10 06:22 fast.log
-rw-r----- 1 root root 1.9M Mar 10 06:22 stats.log
```
Exploratory Data Analysis
Getting the data to the VM

- Index/Parse the Data with Splunk
  - `sudo su - splunk -c "cd /opt/splunk/etc/apps && git clone https://github.com/anthonygtellez/TA-Suricata.git"
  - `/opt/splunk/bin/splunk add index suricata`
  - `/splunk add oneshot /tmp/suricata_malware/eve.json – sourcetype suricata –index suricata`
Exploratory Data Analysis
Getting the data to the VM

Search & Validate
- Index=suricata sourcetype=suricata

Optional*
source=/tmp/suriata_malware/eve.json
Exploratory Data Analysis
Getting the data to the VM

- Index /etc/suricata/rules
  - Index=surirules
  - **Experimental**

https://github.com/anthonygtellez/TA-Suricata_rules.git
Iteration & Experimentation

Nothing is perfect the first time
Iteration & Experimentation
Nothing is perfect the first time

- With both the rules and data in Splunk we can compare signatures and the alerts which fired for the various pcap files.
- This gives us a framework and single pane of glass as an analyst to see what an IDS does and the rules associated with an alert.
Finding which rules fired (the easy way)

- `index=suricata sourcetype="suricata" event_type=alert`  
  `| table alert.signature_id`  
  `| dedup alert.signature_id`  
  `| sort alert.signature_id`

Use as a subsearch against the rules index

- `index=surirules [] | search index=suricata sourcetype="suricata" event_type=alert`  
  `| table alert.signature_id`  
  `| dedup alert.signature_id`  
  `| rename alert.signature_id AS rule_sid`
Finding which rules have events in Splunk (the reverse!)

- index=surirules
  - table rule_sid
  - dedup rule_sid
  - sort rule_sid

Use as a subsearch against the event index

- index=suricata sourcetype="suricata"
  - event_type=alert [[search index=surirules
  - table rule_sid
  - dedup rule_sid
  - rename rule_sid AS alert.signature_id]
The cluster command or patterns tab can be used to cluster rules and events together to identify trends.

- Clustering produces poor patterns when there are less than 5000 events.
- These events can be used to identify general trends about the signatures firing in your environment or bias in the sample data.
- In this case most of the sample malware is firing on Tor node ip addresses.
Operationalizing Insights

How to justify your time to management
This information gives us insights into the types of rules likely to fire and provides the analysts information about how to write a simple signature.

All of them are ip based signatures

- alert ip [95.91.100.46,95.91.1.170,96.126.10 5.219,96.126.117.198,96.126.125.18 7,96.126.96.9,96.126.96.90,96.18.18 2.94,96.235.130.252,96.240.10.123] any -> $HOME_NET any ...
This can be further operationalized by looking at patterns in the data itself.

Remember all of the data in the Suricata index is bad!

An analyst can look at: flow, dns, fileinfo, http, tls, etc and create new signatures based on common patterns they find in the data.

ML & Analytics opportunity?
Operationalizing Insights
How to justify your time to management.

- DNS Nameservers
  - OpenNIC
- TLS & SSL Versions
  - TLS < 1.2, SSLv3
- SSL Issuer
  - CN=www.qpktke29.com/O=qpktke29./C=US
- Cipher Suite?
Operationalizing Insights
How to justify your time to management.

HTTP Content
- How are these users getting infected?
- Are there any rare or common content types?

```
index=suricata event_type=fileinfo
| stats count by http_user_agent
| lookup user_agents http_user_agent
```
Operationalizing Insights
How to justify your time to management.

- **Http Content**
  - Are certain user agents or browsers more common to infection?
  - Is this a specific OS type of infection?
Common trends can be analyzed to write better rules

Is connection to TOR the only giveaway?

Are there things such as query_length, TLS version, etc that are more consistent signatures?

• alert tls any any -> any any (msg:"SURICATA TLS invalid SSLv2 header"; flow:established; app-layer-event:tls.invalid_sslv2_header; flowint:tls.anomaly.count,+,1; classtype:protocol-command-decode; sid:2230000; rev:1;)

What about certain certificates?


Rule Writing
Operationalizing Insights
How to justify your time to management.

► Problems
  • Data becomes unlabeled
  • Lots of different PCAP files get stored into single Splunk index
  • Events overlap difficult to write a specific signature based on lots of samples of one type of malware when multiple malware are added to the same index

► Suggestion (will make pull request)
  • Metadata - Pcap-filename as output path
  • Run each pcap or groups of the same malware in separate eve.logs with labeled paths
  • Alt: use interface to save PCAP filename
    • -This is how Splunk Stream does it
Thank You
Credit
Who I stole stuff from / learned from

- Malware Traffic Analysis, - for the cool labeled data and helping me on my security journey
- Jackson Sie, Vladimir Shcherbakov, David Cuvto + Stream SMEs
- Suricata team for teaching me how to create sockets!
- David Shpritz for the User Agent TA [https://splunkbase.splunk.com/app/1843](https://splunkbase.splunk.com/app/1843)
- Other cool projects for training analysts
  - [https://github.com/threatstream/mhn](https://github.com/threatstream/mhn)
- **Snort SSL TLS inspection**