Hunting and Automation Using Open Source Tools

John Holowczak
Threat Researcher, Threat Analysis Unit
Carbon Black
@skipwich

Brian Baskin
Sr. Threat Researcher, Threat Analysis Unit
Carbon Black
@bbaskin

Carbon Black.

20 June 2019 - FIRST
CB TAU Threat Researchers

John Holowczak

- Development of TAU Research lab, Automation, Binary Analysis, Hunting
- Seasoned security operations analyst

Brian Baskin

- Research, Reverse Engineering, Automation, Hunting
- Formerly with US Defense Cyber Crime Center (DC3) focusing on military network intrusions
# Baselining: What and Why

# Blueprinting your Organization

# Automate the SOC

# Threat Hunting
Baselining: What and Why
Know Your Environment
Know Your Environment

- With Baselineing, turn over every stone; even normal behavior may be abnormal in reality
- When processing data, classify normal behavior and abnormal behavior
  - Certain behaviors can have multiple classifications
- Start your classification buckets large, add detail after each pass
Baselining vs Blueprinting Methods

<table>
<thead>
<tr>
<th>Reactive</th>
<th>Proactive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firehose of data</td>
<td>Blueprint First</td>
</tr>
<tr>
<td>Ingest all data</td>
<td>Create rules for abnormal behavior</td>
</tr>
<tr>
<td>into SIEM</td>
<td>Suffer less False Positives</td>
</tr>
<tr>
<td>Tune False</td>
<td></td>
</tr>
<tr>
<td>Positives Forever</td>
<td></td>
</tr>
</tbody>
</table>
Tools and Procedures

splunk®

ELK Stack! Get it?

Elasticsearch
Logstash
Kibana
Enterprise Blueprinting
osquery

- Open source tool for querying endpoint metadata (at scale) like a database
- Utilizes SQL to expose data via a common interface
- Extensible in a number of languages
  - Add your own query-able data types
• Easy to get data from a number of endpoints at scale
  • Quickly query data using a common language (SQL)
  • Exhaustive list of metadata that is continually growing

• May be difficult to deploy across entire environment
  • Common orchestration tools can help with this (Ansible, Puppet, Chef)!
OSQuery Further Information

• List of schema available at https://osquery.io/schema/

• Some schema offer event information such as process_file_events which includes timestamps with when an event took place
  - Can only get this info if running OSquery in daemon mode, as it is an evented table

• Other file information schema:
  - Signature information
  - Startup items
  - Scheduled tasks
Low Prevalence Executables
Low Prevalence Executables

- One-offs or rare applications
- Care less about the most common running programs
- Classify normal and abnormal for rarities to job functions
Leveraging OSQuery

```sql
osquery> SELECT name, pid, path, start_time FROM processes;
```

<table>
<thead>
<tr>
<th>name</th>
<th>pid</th>
<th>path</th>
<th>start_time</th>
</tr>
</thead>
<tbody>
<tr>
<td>systemd</td>
<td>1</td>
<td>/usr/lib/systemd/systemd</td>
<td>0</td>
</tr>
<tr>
<td>rcu_sched</td>
<td>10</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>migration/18</td>
<td>100</td>
<td>/usr/libexec/udisks2/udisksd</td>
<td>0</td>
</tr>
<tr>
<td>udisksd</td>
<td>1002</td>
<td>/usr/libexec/udisks2/udisksd</td>
<td>15</td>
</tr>
<tr>
<td>ksoftirqd/18</td>
<td>101</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>systemd-logind</td>
<td>1024</td>
<td>/usr/lib/systemd/systemd-logind</td>
<td>16</td>
</tr>
<tr>
<td>gssproxy</td>
<td>1025</td>
<td>/usr/sbin/gssproxy</td>
<td>16</td>
</tr>
<tr>
<td>irqbalance</td>
<td>1026</td>
<td>/usr/sbin/irqbalance</td>
<td>16</td>
</tr>
<tr>
<td>smartd</td>
<td>1028</td>
<td>/usr/sbin/smartd</td>
<td>16</td>
</tr>
<tr>
<td>kworker/18:0H</td>
<td>103</td>
<td>/usr/bin/lsmd</td>
<td>0</td>
</tr>
<tr>
<td>lsmd</td>
<td>1032</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>watchdog/19</td>
<td>104</td>
<td>/usr/sbin/watchdog</td>
<td>0</td>
</tr>
<tr>
<td>alsactl</td>
<td>1040</td>
<td>/usr/sbin/alsactl</td>
<td>16</td>
</tr>
<tr>
<td>migration/19</td>
<td>105</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>kworker/23:2</td>
<td>10555</td>
<td></td>
<td>17627181</td>
</tr>
<tr>
<td>ksoftirqd/19</td>
<td>106</td>
<td>/usr/sbin/alsactl</td>
<td>0</td>
</tr>
<tr>
<td>mcelog</td>
<td>1068</td>
<td>/usr/sbin/mcelog</td>
<td>17627184</td>
</tr>
<tr>
<td>kworker/19:0H</td>
<td>108</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>kworker/16:2</td>
<td>10880</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>watchdog/20</td>
<td>109</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Networking Data

• SNMP (or equivalent) to pull data from your networking devices
• ARP Tables are a great start for network data collection
• Acquire IP and MAC addresses easily
• MAC Addresses are a great way to identify vendors on your network
Common OUIs

https://www.wireshark.org/tools/oui-lookup.html

00:05:69 VMware VMware, Inc.
00:06:1B Notebook Notebook Development Lab. Lenovo Japan Ltd.
00:0C:29 Lenovo Lenovo Mobile Communication Technology Ltd.
00:55:07 LenovoEMC LenovoEMC Products USA, LLC
00:CB:85 Motorola Motorola Mobility LLC, a Lenovo Company
14:1A:A3 Motorola Motorola Mobility LLC, a Lenovo Company
14:30:C6 Motorola Motorola Mobility LLC, a Lenovo Company
14:00:97 DellEMC Dell EMC
00:01:44 DellEMC Dell EMC
00:06:5B Dell Dell Inc.
00:08:74 Dell Dell Inc.
00:0B:DB Dell Dell Inc.
00:0D:56 Dell Dell Inc.
00:0F:1F Dell Dell Inc.
00:11:43 Dell Dell Inc.
00:12:3F Dell Dell Inc.
00:12:48 DellEMC Dell EMC
00:13:72 Dell Dell Inc.
00:14:22 Dell Dell Inc.
00:15:30 DellEMC Dell EMC
00:15:C5 Dell Dell Inc.
Using OSQuery to Enrich Networking Data

• OSQuery is a great tool to grab point-in-time endpoint data to supplement networking data

• Compare NICs and ARP tables on endpoint against Networking equipment ARP tables

• Great way to do full-coverage rogue detection
Getting ARP Data from OSQuery

- Using the `osqueryi` command locally we can test out our queries before running against our environment.
SOC Automation

Easing the task of baselining
Automation Overview

- Running minor, repetitive tasks
- Allows teams freedom to study data
  - Data Collection
  - Aggregation
- Scripts to find badness
  - This is what SIEM’s do!
- Scripts making decisions for us
  - “This machine fell outside baseline, I’ll automatically ban it”
Where do you put your data?

splunk®

ELK Stack!
Get it?

Elasticsearch
Logstash
Kibana

sumologic
Data Collection

[Diagram showing the process from Python, GitHub, Stack Overflow to a database]
Querying Data

• Jupyter Notebook or R to automate studies
  - Programming heavy
  - Steeper learning curve

• ELK for querying
  - May take longer to set up
  - Large community support
Docker

• Public Docker for simple ELK stacks exist

• Require configuring data inputs (more involved)

• Seamlessly integrate OSQuery into Logstash using Filebeat (part of ELK)

• Easy-to-follow guide: https://elk-docker.readthedocs.io/
Filebeat

- Log forwarding service, part of ELK stack
- Has built-in templates for transforming OSQuery data into an easily-digestible format.
- OSQuery also has built in support for pushing to LogStash
Configuring OSQuery for Scheduled Queries

```json
{
    "options": {
        "host_identifier": "hostname",
        "schedule_splay_percent": 10
    },
    "schedule": {
        "arp_cache": {
            "query": "SELECT * FROM arp_cache;",
            "interval": 10
        }
    }
}
```
Next Steps

• Forward OSQuery (using Filebeat) to Logstash, start hunting with Kibana
Threat Hunting

Tying it all together
Using Statistical Analysis for Threat Hunting

- **Baseline**: Determine the “normal” dataset
- **Filter**: Eliminate baseline as noise
- **Identify**: Investigate anomalies & statistical deviations
Analyzing Data

In [6]:

sns.set(style="white", palette="muted", color_codes=True)
f, axes = plt.subplots(1, 1, figsize=(7, 7), sharex=True)
sns.despine(left=True)
sns.distplot(cs, hist=False, color="g", kde_kws={"shade": True})
plt.set(axes, yticks=[])  
plt.tight_layout()
Hunting Methodologies

- Back to the basics: Now time to look for the abnormal
- Search across environments for behavior and static IOC’s
- Least prevalent occurrences tend to be most abnormal
## MAC Addresses – Uncommon Environment OUIs

<table>
<thead>
<tr>
<th>10</th>
<th>00:50:56</th>
<th>10</th>
<th>00:50:56</th>
<th>10</th>
<th>00:50:56</th>
<th>10</th>
<th>00:50:56</th>
<th>10</th>
<th>00:50:56</th>
<th>10</th>
<th>52:54:00</th>
<th>10</th>
<th>00:50:56</th>
<th>10</th>
<th>00:50:56</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Vmware</td>
<td></td>
<td>Vmware</td>
<td></td>
<td>Vmware</td>
<td></td>
<td>Vmware</td>
<td></td>
<td>Vmware</td>
<td></td>
<td>Vmware</td>
<td></td>
<td>Vmware</td>
<td></td>
<td>Vmware</td>
</tr>
<tr>
<td></td>
<td>VMware Inc.</td>
<td></td>
<td>VMware Inc.</td>
<td></td>
<td>VMware Inc.</td>
<td></td>
<td>VMware Inc.</td>
<td></td>
<td>VMware Inc.</td>
<td></td>
<td>VMware Inc.</td>
<td></td>
<td>VMware Inc.</td>
<td></td>
<td>VMware Inc.</td>
</tr>
</tbody>
</table>
# OSQuery Hunting

```sql
osquery> SELECT name, pid, path, start_time FROM processes;
```

<table>
<thead>
<tr>
<th>name</th>
<th>pid</th>
<th>path</th>
<th>start_time</th>
</tr>
</thead>
<tbody>
<tr>
<td>systemd</td>
<td>1</td>
<td>/usr/lib/systemd/systemd</td>
<td>0</td>
</tr>
<tr>
<td>rcu_sched</td>
<td>10</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>migration/18</td>
<td>100</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>udisksd</td>
<td>1002</td>
<td>/usr/libexec/udisks2/udisksd</td>
<td>15</td>
</tr>
<tr>
<td>ksoftirqd/18</td>
<td>101</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>systemd-logind</td>
<td>1024</td>
<td>/usr/lib/systemd/systemd-logind</td>
<td>16</td>
</tr>
<tr>
<td>gssproxy</td>
<td>1025</td>
<td>/usr/sbin/gssproxy</td>
<td>16</td>
</tr>
<tr>
<td>irqbalance</td>
<td>1026</td>
<td>/usr/sbin/irqbalance</td>
<td>16</td>
</tr>
<tr>
<td>smartd</td>
<td>1028</td>
<td>/usr/sbin/smartd</td>
<td>16</td>
</tr>
<tr>
<td>kworker/18:0H</td>
<td>103</td>
<td>/usr/bin/lsmd</td>
<td>0</td>
</tr>
<tr>
<td>lsmd</td>
<td>1032</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>watchdog/19</td>
<td>104</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>alsactl</td>
<td>1040</td>
<td>/usr/sbin/alsactl</td>
<td>16</td>
</tr>
<tr>
<td>migration/19</td>
<td>105</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>kworker/23:2</td>
<td>10555</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>ksoftirqd/19</td>
<td>106</td>
<td>/usr/sbin/ksoftirqd</td>
<td>17</td>
</tr>
<tr>
<td>mcelog</td>
<td>1068</td>
<td>/usr/sbin/mcelog</td>
<td>17</td>
</tr>
<tr>
<td>kworker/19:0H</td>
<td>108</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>kworker/16:2</td>
<td>10880</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>watchdog/20</td>
<td>109</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

© 2019 Carbon Black Incorporated. All rights reserved. Carbon Black and CB Predictive Security Cloud are registered trademarks or trademarks of Carbon Black, Inc. in the United States and/or other jurisdictions. All other trademarks and registered trademarks are the property of their respective owners.
OSQuery Hunting

SELECT name FROM processes WHERE start_time < 100;

7189 csrss.exe
7189 lsass.exe
7189 conhost.exe
...
2348 dllhost.exe
103 firstboot.cmd
45 FlashPlayerUpdateService.exe
8 psexesvc.exe
3 rundll.exe
1 conh0st.exe
Prevalence of Executables

- Can you:
  - Identify abnormal software running on fewest endpoints?
  - Identify executables that are widespread but in unusual places?

- Yes!
  - Extract data on binaries from osquery
  - Combine into CSVs and perform text magic
Filtering Data

- Expressions to hunt for unusual indicators
  - Files that have a single character filename: \\
  - Files running one-folder deep from volume root: (:\\[a-zA-Z0-9]{1,12}\[a-zA-Z0-9]*\
  - Files run directly from Windows folder: (:\\windows\\{1,15},)
  - Files with unusual extensions: (\bin,\dat,\log,\gif,\txt,\jpg,\rar,\sql,)
Mass Searching

• One-character file names:
  6 c:\tdm-gcc-64_4.9.2\work\a.exe
  1 c:\acccb\agusta\y.bat
  1 c:\users\jsmith\appdata\local\microsoft\windows\temporary internet files\content.ie5\4unu162n\..exe
  1 sysvol\users\z9service\downloads\q.exe
  1 sysvol\program files (x86)\k2 for sharepoint 2013\z.bat

• Low prevalence in Windows Folder
  22 c:\windows\psexesvc.exe
  1 c:\windows\system32\oem\firstboot.cmd
  1 sysvol\windows\system32\dsget.exe
  1 c:\windows\system32\hpbtpro.exe
  1 c:\windows\system32\scardsvr.exe
A Story of Two Executables (PLink)
Happy Hunting!
John Holowczak
@skipwich
jholowczak@carbonblack.com

Brian Baskin
@bbaskin
bbaskin@carbonblack.com