MOBIA

MOBIA is a business technology integrator with over thirty years of experience and 500+ employees across Canada and USA. Our talented bench of technical engineers and trusted advisors deliver process improvements and business transformations within our core pillars of Cloud, Infrastructure, Software Development, Cybersecurity and Broadband & Wireless Services.

Our inside-out approach allows us to understand business challenges from deep within the organization, mapping the impact as it ripples outward. This insight enables us to create future-proof solutions that maximize results and repeatedly exceed our client's expectations.

# Adversary Simulation Workshop

## Lab Guide:
## Events and Detection

# Objective:

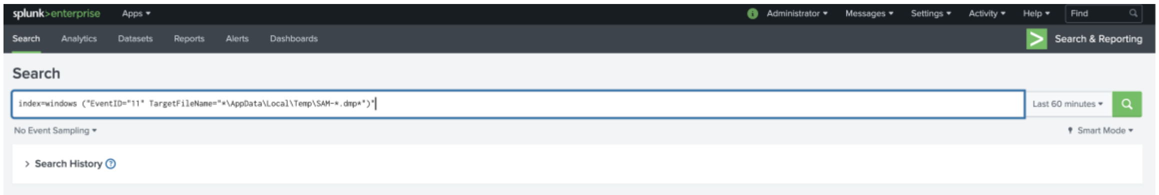
In this lab we will use the search we created in the previous lab (sigma rules) to attempt to fix evidence of the activity.

# Instructions:

Browse to your student instance of Splunk. The url and credentials will be provided by your instructor.

After successful login you will be dropped into the default Splunk search app. In the search bar enter the search string from the previous lab making sure to provide a Splunk index. Notice we've updated the TargetFileName to be a more generic. This way we'll be able to use the procdump test included in T1003.001.

```
(EventID=11 TargetFilename="*\Temp\*.dmp")
```



```
index=windows ("EventID="11" TargetFileName="*\AppData\Local\Temp\SAM-*.dmp*")"
```

**Note**:
For the lab environment all windows logs including Sysmon are all found in the index "windows".

Run the search for the past 60 minutes. There should be no results and no errors returned.

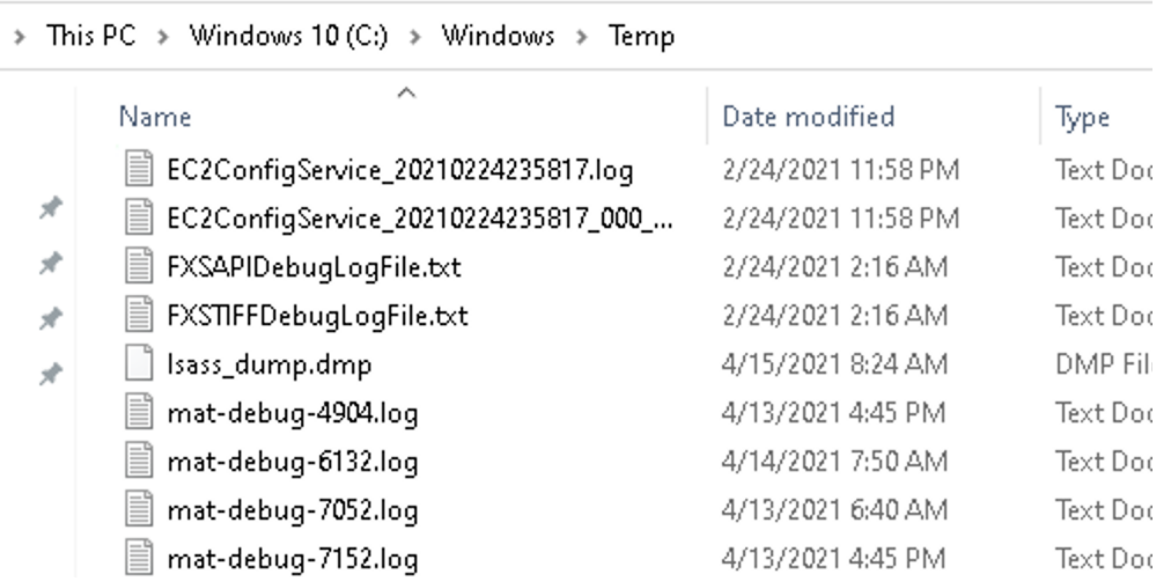
## Execute Procdump Test

Next we'll need to generate some data on our student lab machine. There isn't a test case that exists for PWDump but in T1003.001 test #2 will allow us to generate an LSASS dump event similar for our test.

On the student machine run the following command to execution the test and get the prereqs:

```
Invoke-AtomicTest T1003.001 -GetPrereqs -TestNumbers 2
```

Under C:\Windows\Temp\ you should have a file named lsass_dump.dmp

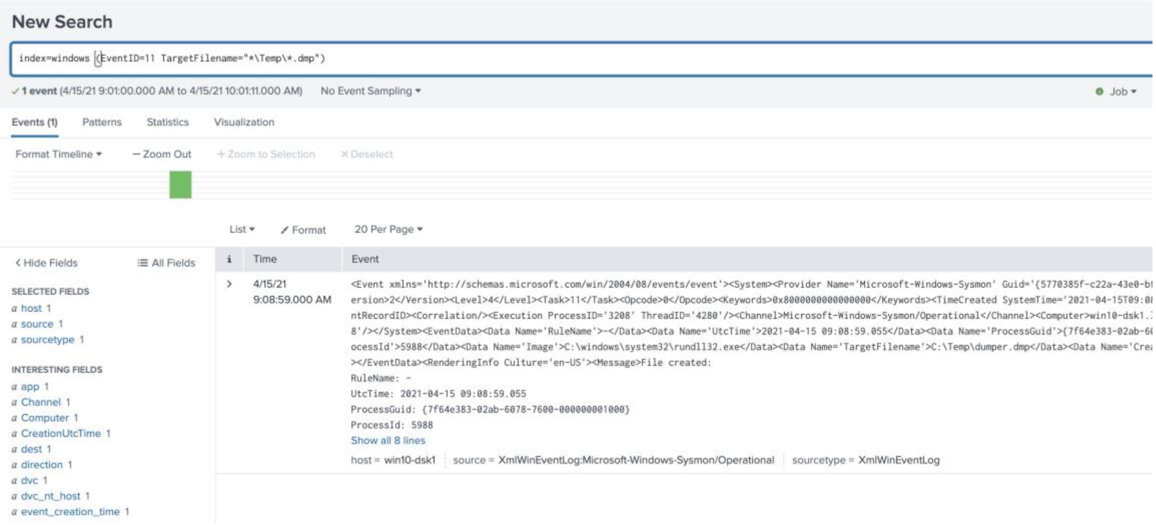


› This PC › Windows 10 (C:) › Windows › Temp

| Name | Date modified | Type |
|---|---|---|
| EC2ConfigService_20210224235817.log | 2/24/2021 11:58 PM | Text Doc |
| EC2ConfigService_20210224235817_000_... | 2/24/2021 11:58 PM | Text Doc |
| FXSAPIDebugLogFile.txt | 2/24/2021 2:16 AM | Text Doc |
| FXSTIFFDebugLogFile.txt | 2/24/2021 2:16 AM | Text Doc |
| lsass_dump.dmp | 4/15/2021 8:24 AM | DMP Fil |
| mat-debug-4904.log | 4/13/2021 4:45 PM | Text Doc |
| mat-debug-6132.log | 4/14/2021 7:50 AM | Text Doc |
| mat-debug-7052.log | 4/13/2021 6:40 AM | Text Doc |
| mat-debug-7152.log | 4/13/2021 4:45 PM | Text Doc |

## Search for Events

Log back into the Splunk server and re-run your initial search. You should see a number of events returned:



New Search

```
index=windows (EventID=11 TargetFilename="*\Temp\*.dmp")
```

✓ 1 event (4/15/21 9:01:00.000 AM to 4/15/21 10:01:11.000 AM)

SELECTED FIELDS
- a host 1
- a source 1
- a sourcetype 1

INTERESTING FIELDS
- a app 1
- a Channel 1
- a Computer 1
- a CreationUtcTime 1
- a dest 1
- a direction 1
- a dvc 1
- a dvc_nt_host 1
- a event_creation_time 1

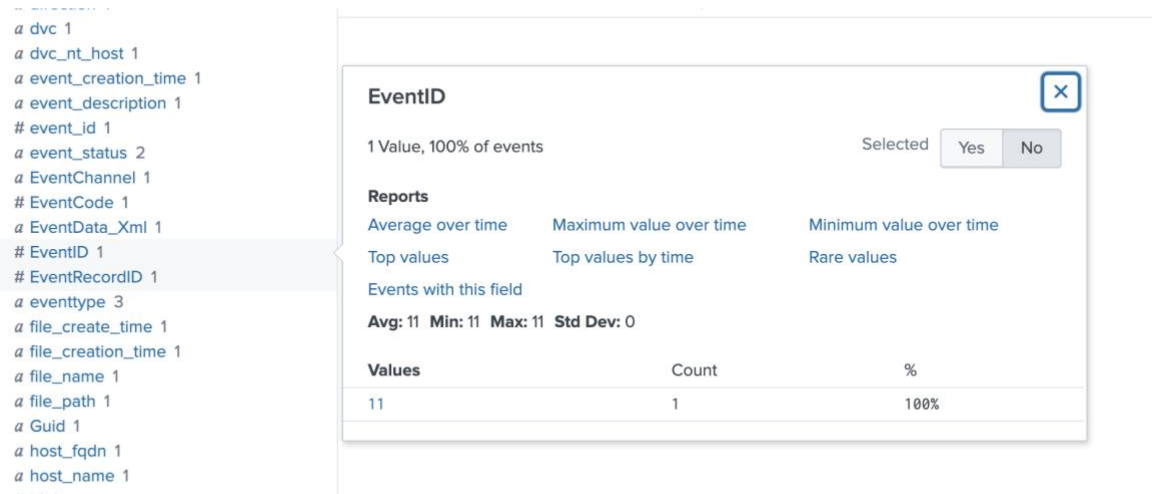
| i | Time | Event |
|---|---|---|
| › | 4/15/21<br>9:08:59.000 AM |  |

```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Sysmon' Guid='{5770385f-c22a-43e0-bf
ersion>2</Version><Level>4</Level><Task>11</Task><Opcode>0</Opcode><Keywords>0x8000000000000000</Keywords><TimeCreated SystemTime='2021-04-15T09:0
ntRecordID><Correlation/><Execution ProcessID='3208' ThreadID='4280'/><Channel>Microsoft-Windows-Sysmon/Operational</Channel><Computer>win10-dsk1.
8'/></System><EventData><Data Name='RuleName'>-</Data><Data Name='UtcTime'>2021-04-15 09:08:59.055</Data><Data Name='ProcessGuid'>{7f64e383-02ab-6
ocessId'>5988</Data><Data Name='Image'>C:\windows\system32\rundll32.exe</Data><Data Name='TargetFilename'>C:\Temp\dumper.dmp</Data><Data Name='Cre
></EventData><RenderingInfo Culture='en-US'><Message>File created:
RuleName: -
UtcTime: 2021-04-15 09:08:59.055
ProcessGuid: {7f64e383-02ab-6078-7600-000000001000}
ProcessId: 5988
```

Show all 8 lines

host = win10-dsk1 | source = XmlWinEventLog:Microsoft-Windows-Sysmon/Operational | sourcetype = XmlWinEventLog

Look down the right hand side for all of the extracted fields. Locate the two fields we used in our search TargetFilename and EventID.



a dvc 1
a dvc_nt_host 1
a event_creation_time 1
a event_description 1
# event_id 1
a event_status 2
a EventChannel 1
# EventCode 1
a EventData_Xml 1
# EventID 1
# EventRecordID 1
a eventtype 3
a file_create_time 1
a file_creation_time 1
a file_name 1
a file_path 1
a Guid 1
a host_fqdn 1
a host_name 1

**EventID**

1 Value, 100% of events

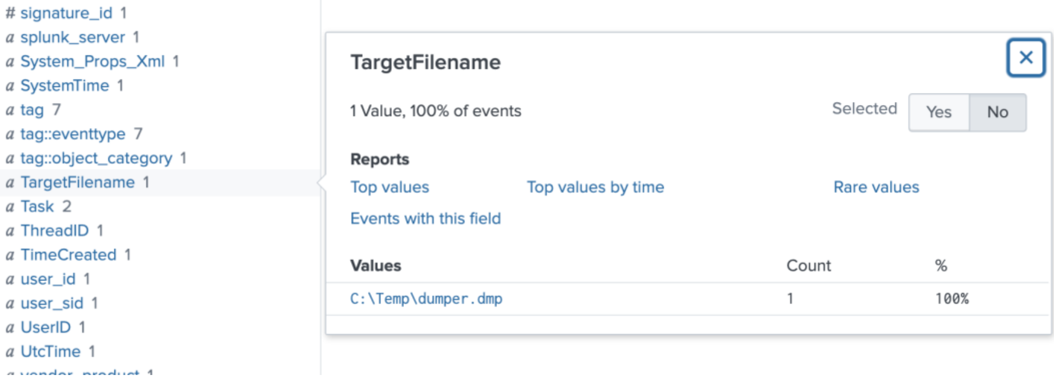
Selected Yes No

**Reports**

Average over time · Maximum value over time · Minimum value over time
Top values · Top values by time · Rare values
Events with this field

**Avg:** 11 **Min:** 11 **Max:** 11 **Std Dev:** 0

| Values | Count | % |
|---|---|---|
| 11 | 1 | 100% |



# signature_id 1
a splunk_server 1
a System_Props_Xml 1
a SystemTime 1
a tag 7
a tag::eventtype 7
a tag::object_category 1
a TargetFilename 1
a Task 2
a ThreadID 1
a TimeCreated 1
a user_id 1
a user_sid 1
a UserID 1
a UtcTime 1

**TargetFilename**

1 Value, 100% of events

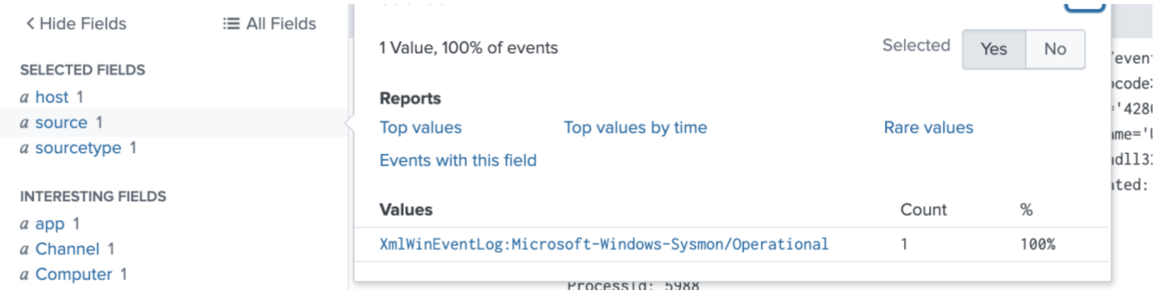
Selected Yes No

**Reports**

Top values · Top values by time · Rare values
Events with this field

| Values | Count | % |
|---|---|---|
| C:\Temp\dumper.dmp | 1 | 100% |

The logs that we've discovered the event in are Sysmon events. We can see this if we look at the source and sourcetype value:



< Hide Fields ☰ All Fields

SELECTED FIELDS
a host 1
a source 1
a sourcetype 1

INTERESTING FIELDS
a app 1
a Channel 1
a Computer 1

1 Value, 100% of events

Selected Yes No

**Reports**

Top values · Top values by time · Rare values
Events with this field

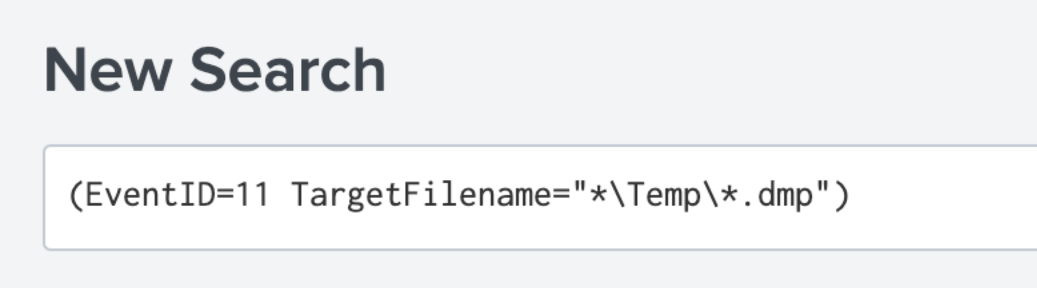
| Values | Count | % |
|---|---|---|
| XmlWinEventLog:Microsoft-Windows-Sysmon/Operational | 1 | 100% |

Now run the search again but remove the filter for the TargetFilename field. Replace that with a count by statement as follows:

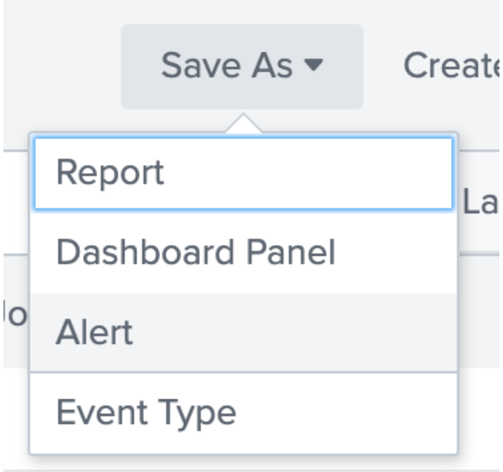
```
EventID=11
| stats count by TargetFilename
```

## Actions

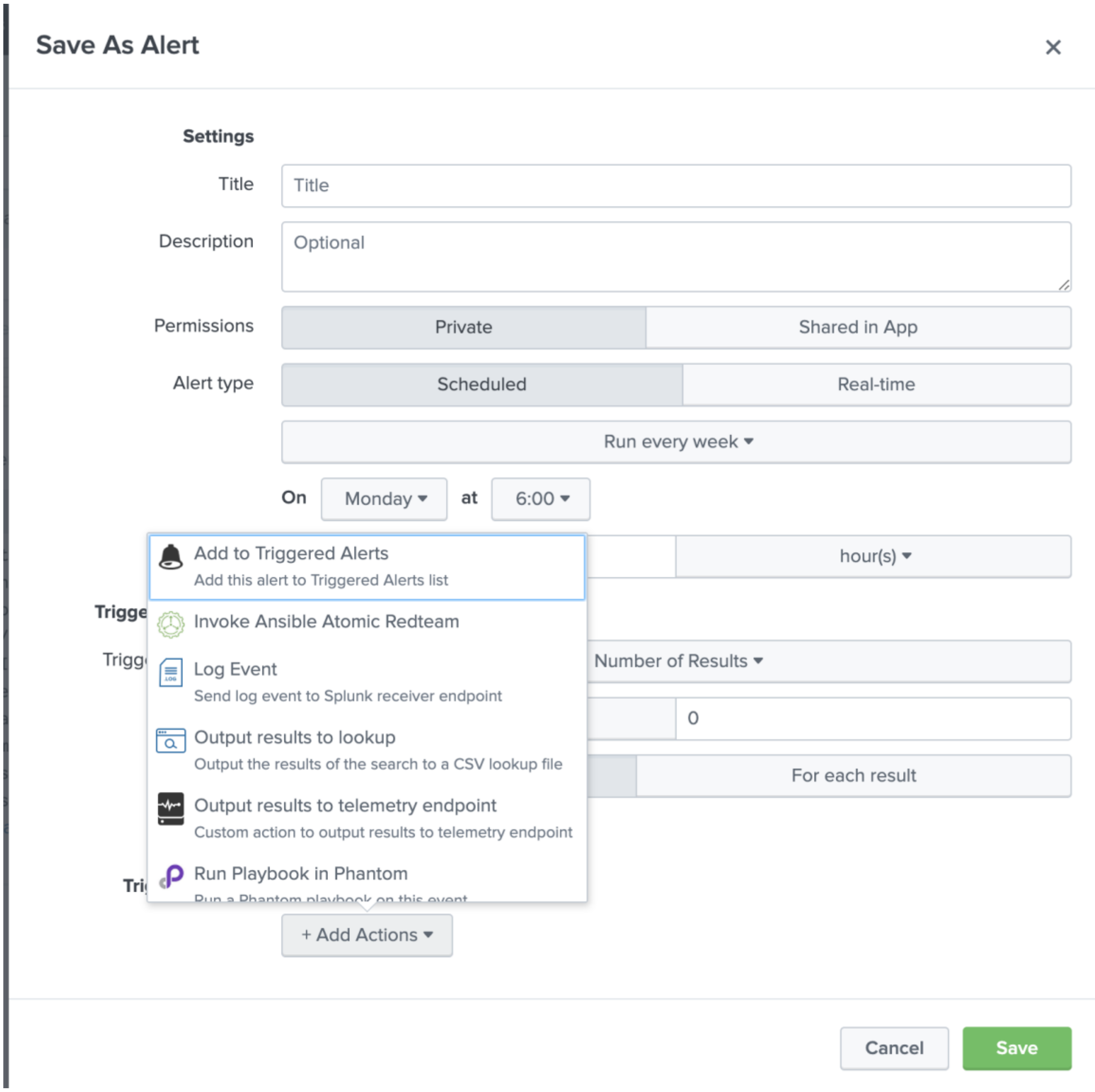
Run the original search you used to locate the procdump file. Then choose Save As



```
(EventID=11 TargetFilename="*\Temp\*.dmp")
```



Click the “+ Add Actions” button to see a list of possible outputs for this alert.



Save As Alert ×

Settings

Title: Title

Description: Optional

Permissions: Private | Shared in App

Alert type: Scheduled | Real-time

Run every week ▾

On Monday ▾ at 6:00 ▾

hour(s) ▾

Trigge

Trigg Number of Results ▾

0

For each result

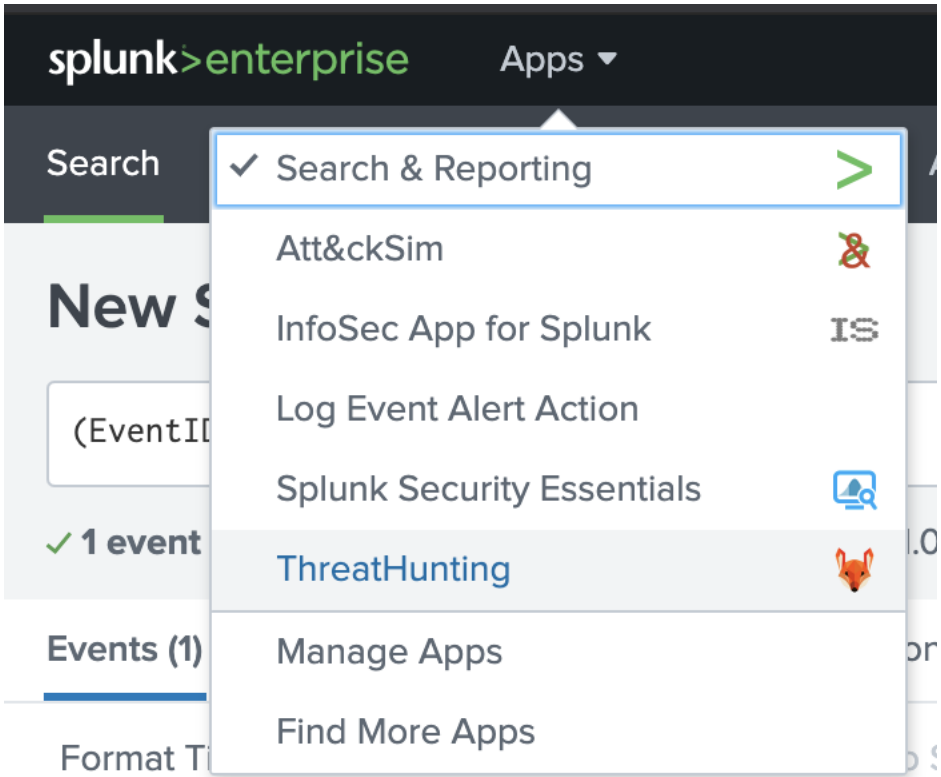
Tri

Add to Triggered Alerts
Add this alert to Triggered Alerts list
Invoke Ansible Atomic Redteam
Log Event
Send log event to Splunk receiver endpoint
Output results to lookup
Output the results of the search to a CSV lookup file
Output results to telemetry endpoint
Custom action to output results to telemetry endpoint
Run Playbook in Phantom
Run a Phantom playbook on this event
+ Add Actions ▾
Cancel
Save

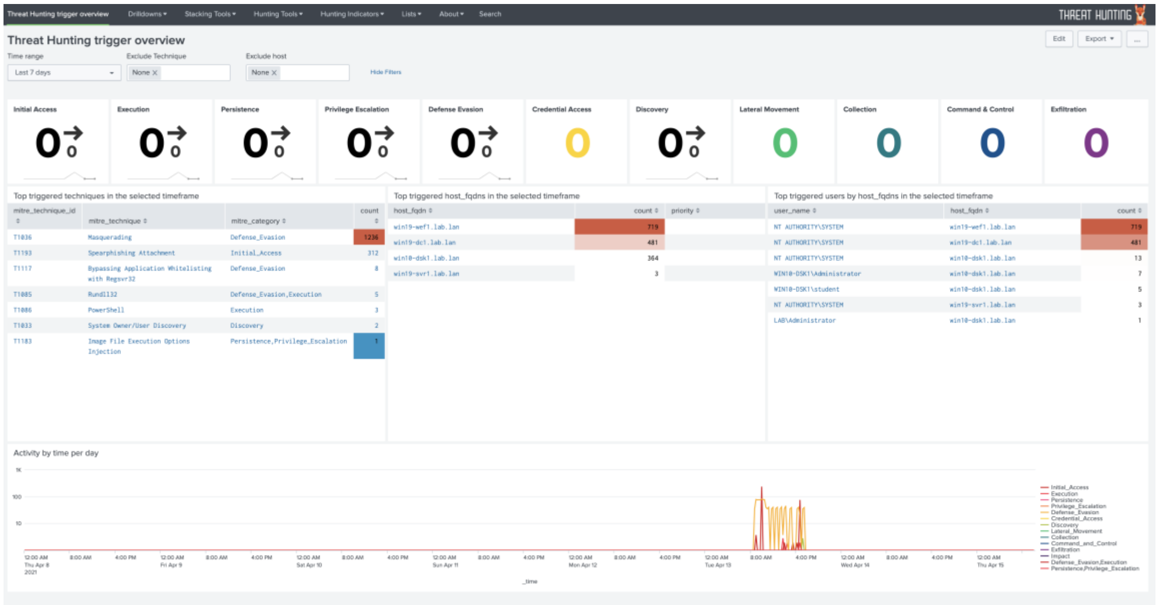
One common technique for alerting on logs that you want to tag as Mitre (or anything else for that matter) is to rewrite the event back to the SIEM. This makes it easier for us to search historically through all of our alerts.

## ThreatHunting App

The threat hunting app has been installed in the lab environment. There is some sample data there to view it in action. Open the app by choosing Threat Hunting from the apps menu:



There are a large number of prebuilt searches based on Sigma in this app.



Top triggered techniques in the selected timeframe

| mitre_technique_id | mitre_technique | mitre_category | count |
|---|---|---|---|
| T1036 | Masquerading | Defense_Evasion | 1236 |
| T1193 | Spearphishing Attachment | Initial_Access | 312 |
| T1117 | Bypassing Application Whitelisting with Regsvr32 | Defense_Evasion | 8 |
| T1085 | Rundll32 | Defense_Evasion,Execution | 5 |
| T1086 | PowerShell | Execution | 3 |
| T1033 | System Owner/User Discovery | Discovery | 2 |
| T1183 | Image File Execution Options Injection | Persistence,Privilege_Escalation | 1 |

Top triggered host_fqdns in the selected timeframe

| host_fqdn | count | priority |
|---|---|---|
| win19-wef1.lab.lan | 719 | |
| win19-dc1.lab.lan | 481 | |
| win10-dsk1.lab.lan | 364 | |
| win19-svr1.lab.lan | 3 | |

Top triggered users by host_fqdns in the selected timeframe

| user_name | host_fqdn | count |
|---|---|---|
| NT AUTHORITY\SYSTEM | win19-wef1.lab.lan | 719 |
| NT AUTHORITY\SYSTEM | win19-dc1.lab.lan | 481 |
| NT AUTHORITY\SYSTEM | win10-dsk1.lab.lan | 13 |
| WIN10-DSK1\Administrator | win10-dsk1.lab.lan | 7 |
| WIN10-DSK1\student | win10-dsk1.lab.lan | 5 |
| NT AUTHORITY\SYSTEM | win19-svr1.lab.lan | 3 |
| LAB\Administrator | win10-dsk1.lab.lan | 1 |

End of Lab