How to Setup Machine Learning and Data Frame Analytic Jobs

Using Elastic Cloud

Produced for CISA by Team Anomalous: Sanjeet Bagga

Background Investigation on APT 29 Dataset

After analyzing the dataset, we found the following activity to be potentially malicious:

Gpt.ini, which is a security configuration file that can be modified into providing privilege escalation. Gpt template overwrites existing policy and can provide admin access onto all boxes or domains if authentication is being done remotely. Gpt file was modified in the NEWYORK Domain, which could provide security policies for Scranton/Nashua. Psexec runs a modified python payload that is copied over from a malicious webserver (determined to be WebDav service). PSExec opens 4 pipes: one for psexec.exe, one for stdin, one for stdout, one for stderr. LSARPC lookupnames3 has been found to enumerate user accounts, and specifically looks for “Domain Admins” and “pbeesly”.

In general:

Stepping through our background investigation, we determined that we would analyze modification of security policies, as they could be malicious. We were also interested in the instances of python.exe and the webdav server, as we determined that the SeaDuke payloads operated through a modified python.exe payload that was connecting over a malicious webserver. If the webserver is not approved and not needed, it should not be generating network traffic. Finally, instances of the LSARPC service (or any Local Security Authority protocol) to enumerate administrative users, should be flagged to ensure that unauthorized access is not given.

Configure Logstash to Ingest Dataset

For more information on ingesting datasets through Logstash, please follow the “How to Configure Logstash to send local data to Elastic Cloud” document.

Please note that the team used both Mordor APT 29 Day1 and Day2 datasets, which were combined through Logstash’s ingestion process.

Determining Important Fields and Features

We determined that because of the irregular behavior of python.exe and the presence of an unverified webdav client, these would be what we would be looking for. See Sanjeet’s and John’s presentations on the datasets.

We also looked at some features in other logged entries that were suspicious activities. These are some examples of logged entries we looked at:

> Network connection detected: RuleName: - UtcTime: 2020-05-02 03:16:54.700 ProcessGuid: {5aa8ec29-e5b8-5eac-7903-000000000400} ProcessId: 2172 Image: C:\Windows\Temp\python.exe User: DMEVALS\pbeesly Protocol: tcp Initiated: true SourceIsIpv6: false SourceIp: 10.0.1.6 SourceHostname: - SourcePort: XXXX SourcePortName: - DestinationIsIpv6: false DestinationIp: 192.168.0.4 DestinationHostname: - DestinationPort: 8443 DestinationPortName: -

> Registry value set: RuleName: - EventType: SetValue UtcTime: 2020-05-02 03:20:42.383 ProcessGuid: {47ab858c-e6ad-5eac-0b00-000000000500} ProcessId: 736 Image: C:\windows\system32\services.exe TargetObject: HKU\.DEFAULT\Software\Classes\Local Settings\MuiCache\4\52C64B7E\@%systemroot%\system32\webclnt.dll,-104 Details: WebDav Client Redirector Drive

> Image:C:\Program Files\SysinternalsSuite\PsExec64.exe @timestamp:May 1, 2020 @ 23:14:43.388 @version:1 @version.keyword:1 AccountName:SYSTEM AccountName.keyword:SYSTEM AccountType:User AccountType.keyword:User Channel:Microsoft-Windows-Sysmon/Operational Channel.keyword:Microsoft-Windows-Sysmon/Operational Company:Microsoft Corporation Company.keyword:Microsoft Corporation Description:Web DAV Client DLL Description.keyword:Web DAV Client DLL Domain:NT AUTHORITY Domain.keyword:NT AUTHORITY EventID:7 EventReceivedTime:2020-05-01 23:14:43 EventReceivedTime.keyword:2020-05-01 23:14:43 EventTime:2020-05-01 23:14:42 EventTime.keyword:2020-05-01 23:14:42 EventType:INFO EventType.keyword:INFO ExecutionProcessID:3,484 FileVersion:10.0.18362.1 (WinBuild.160101.0800) FileVersion.keyword:10.0.18362.1 (WinBuild.160101.0800)
>
> ImageLoaded C:\Windows\System32\davclnt.dll

From this information, we decided to look at the following data points to determine malicious data. Our background research allowed us to understand that python and webclient were probably malicious, and thus we isolated instances of those.

Image: python.exe

User: DMEVALS\pbeesly

Details: WebDav Client Redirector Driver

TargetObject: *webclnt.dll

Event Rate

We used the corresponding features (Image, TargetObject, Details, user) to determine anomalistic references separately, and used queries to filter out for the specific ones if needed.

We found these fields by using Kibana's Discover module, which was instrumental in helping us find anomalies.

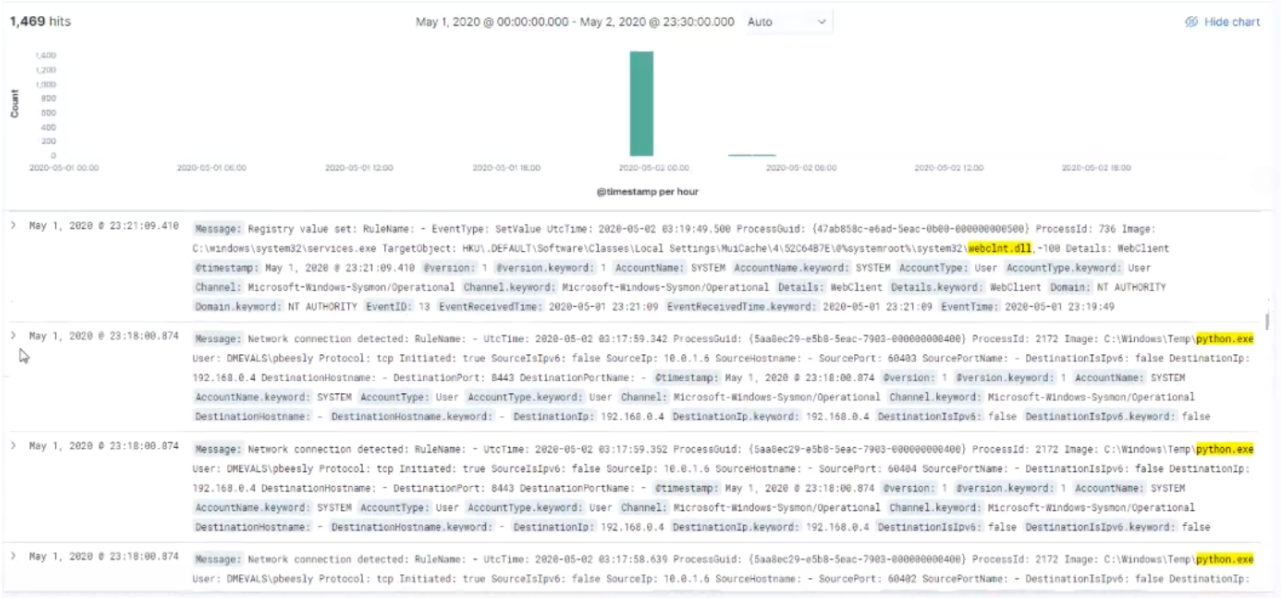


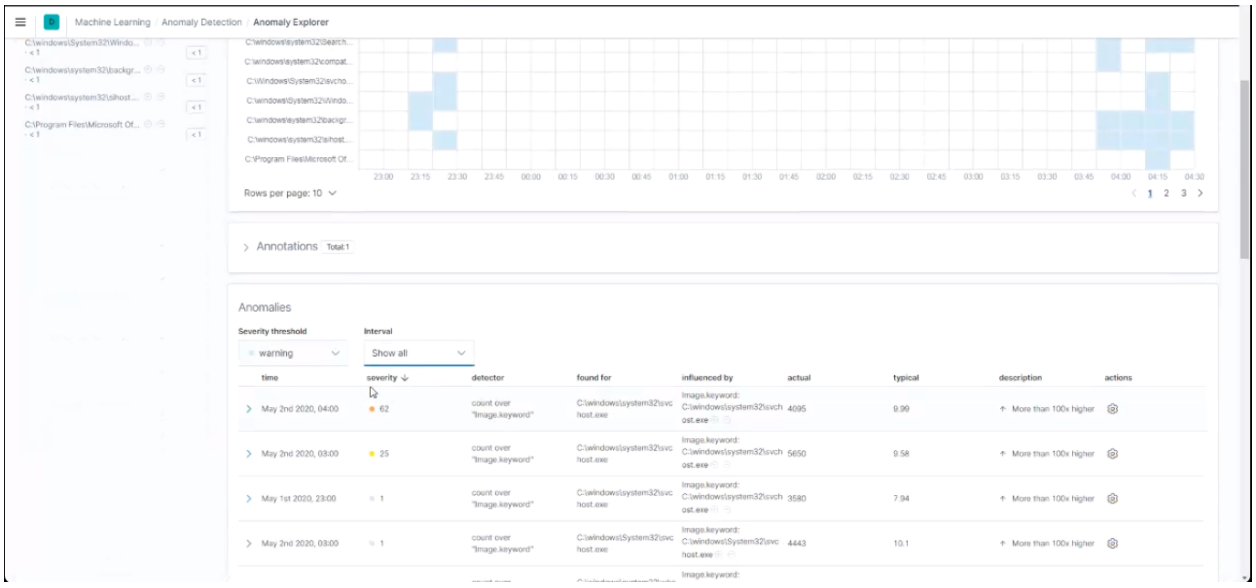
Figure 1: Example of Discover Module with KQL < Message : *webclnt.dll or Message : *python.exe or Message : *psexec64.exe >

Creating ML Jobs for Anomaly Detection

Population:

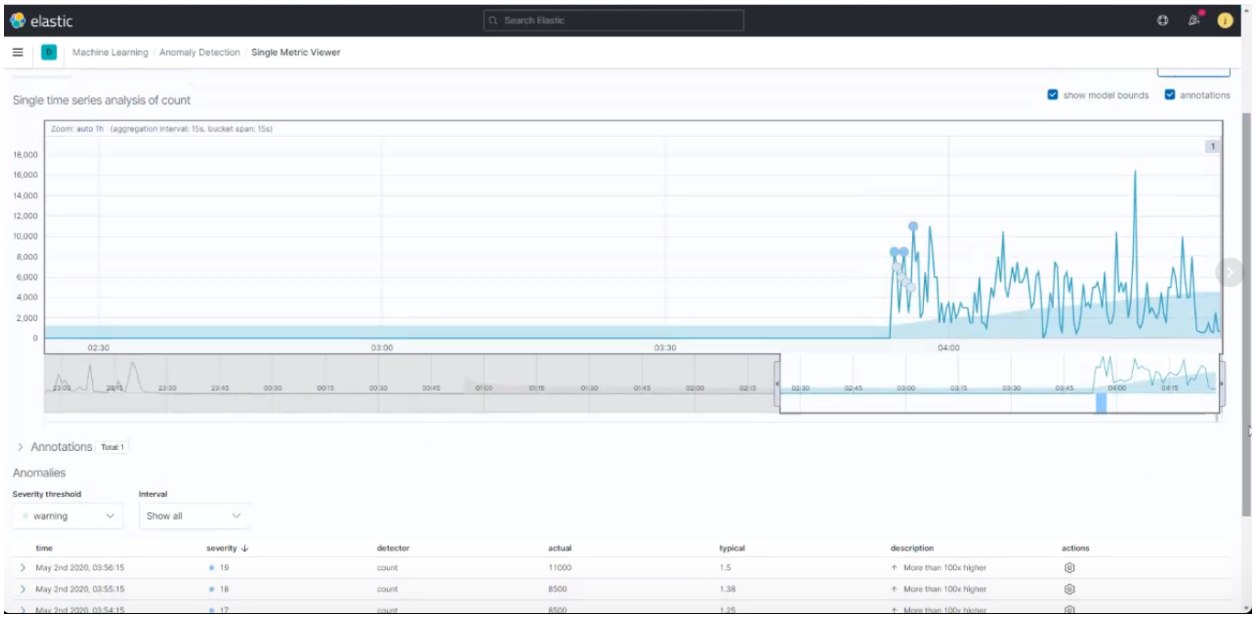
1. Go to Elastic/Machine Learning/Anomaly Detection

2. Click Create Job
3. Select logstashindex dataset
4. Click Population
5. Use full data
6. Click Next
7. Pick fields that were identified above (Image/TargetObject/etc)
8. Bucket Span 15s
9. Click Next all the way through



<u>Single Metric:</u>

1. Go to Elastic/Machine Learning/Anomaly Detection
2. Click Create Job
3. Select logstashindex dataset
4. Click Single Metric
5. Use full data
6. Click Next
7. Pick EventRate as the field
8. Bucket Span 15s
9. Click Next all the way through



Advanced:

1. Go to Elastic/Machine Learning/Anomaly Detection
2. Click Create Job
3. Select logstashindex dataset
4. Click Advanced
5. Click Detectors
6. Function: count
7. Use fields identified above (CountBy and PartitionField TargetObject/Image)
8. Influencers (TargetObject and Image)
9. Bucket Span 15s
10. Click Next all the way through

Detectors

count by "Image.keyword" over "ImageLoaded.keyword"

Job ID
adv-image

Job description
No description provided

Groups
No groups selected

Bucket span
15s

Influencers
Image.keyword, ImageLoaded.keyword, TargetObject.keyword

Enable model plot
False

Use dedicated index
False

Model memory limit
17MB

Datafeed configuration

Elasticsearch query

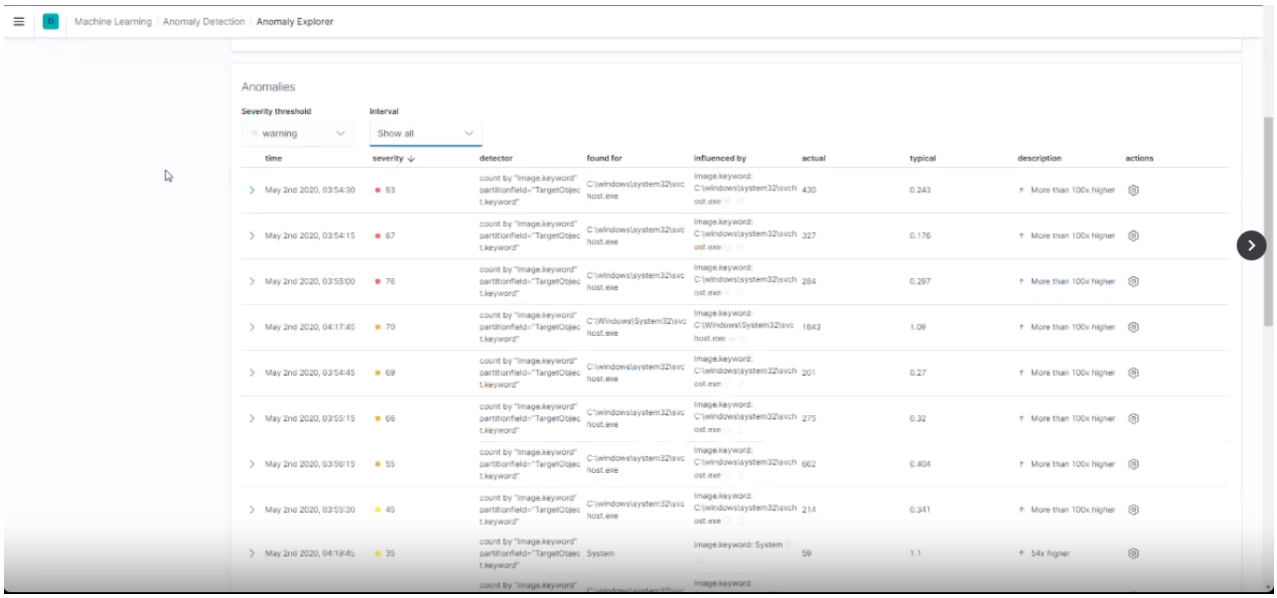
```
{
  "bool": {
    "must": [
      {
        "match_all": {}
      }
    ]
  }
}
```

Time field
@timestamp

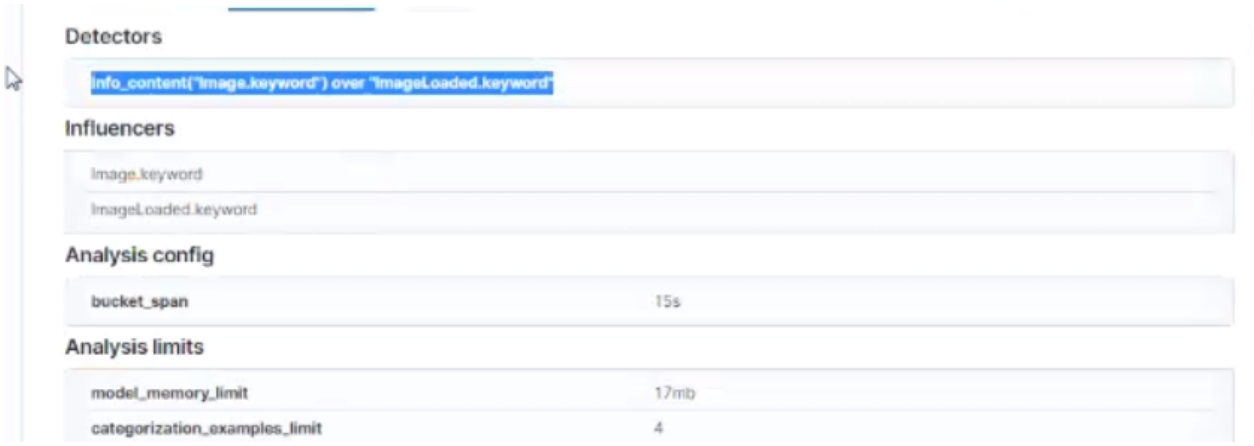
Query delay
60s (default)

Frequency
60s (default)

Scroll size
1000 (default)



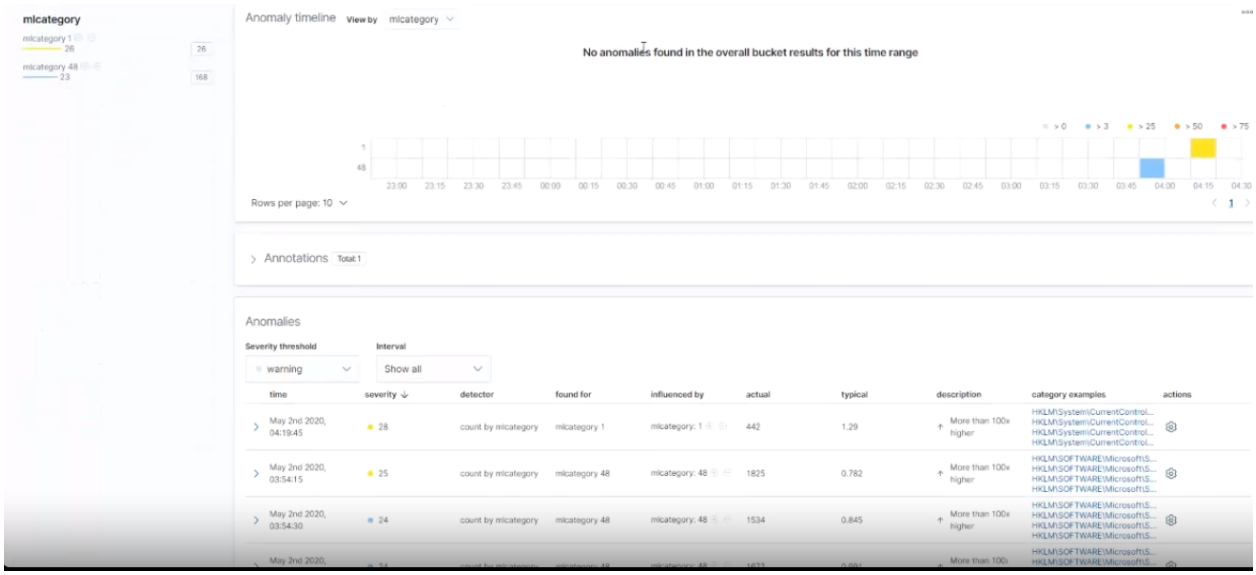
Advanced jobs use a lot of memory, so increasing the size per zone of your Elasticsearch deployment may be necessary (maximum for a trial is 120 GB storage/4 GB RAM)

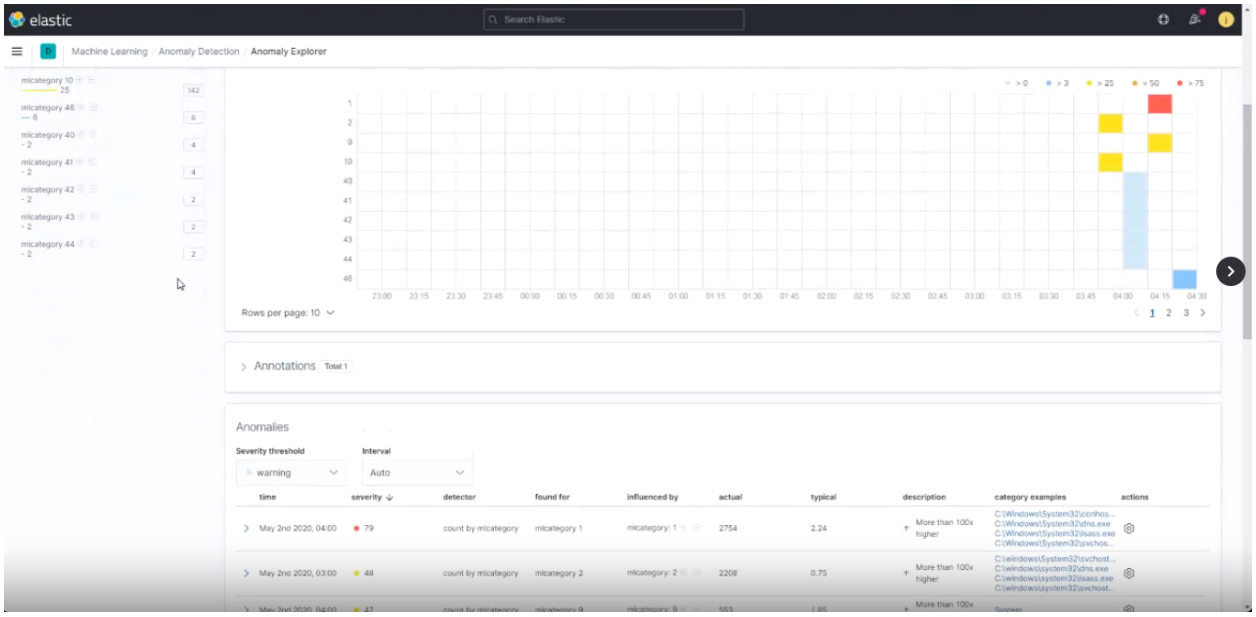


Categorization:

1. Go to Elastic/Machine Learning/Anomaly Detection
2. Click Create Job
3. Select logstashindex dataset
4. Click Categorization
5. Use full data
6. Click Next
7. Categorization by “Count” and fields that were identified above (Image/TargetObject/etc)
8. Bucket Span 15s
9. Click Next all the way through

All of these resulted in some anomalistic data points to put into Kibana
The most important found was from Categorization from the TargetObject & Image features.





ML Jobs for Data Frame Analytics

Data Frame Analytics jobs were used in conjunction with the Anomaly Detection jobs in order to verify and validate whether data found was truly anomalistic. This is due to the ability of the Data Frame Analytics jobs to analyze the criticality of specified features and determine their "Probability Score", along with the reasoning behind it. In testing, it was found that there was a correlation between the anomalies detected in the Anomaly Detection jobs and the Data Frame Analytic job results.

To determine known malicious data and the criticality of outliers, the Data Frame Analytics machine learning job was utilized.

<u>To Create Data Frame Analytics job:</u>

1. Go to Elastic/Machine Learning/Data Frame Analytics
2. Create Job
3. Select "Outlier Detection"
4. Query <feature>:<malicous type>
    a. Ex: Image: python.exe
5. Click Continue
6. Click Continue
7. Create Job ID
8. Click Create

You will be able to sort/filter out per feature of the dataset (Image, TargetObject, EventID) as well as a machine learning outlier score. It is a manual process to determine which columns should be shown in the final job.