Zeek Log Analysis Using ElasticSearch

Key Contributors:
Michael Garcia
Daniel Garza

UNIVERSITY OF THE
INCARNATE WORD®

29 November 2022

Table of Contents

Page

Executive Summary

In this report, we parsed a known malicious pcap file into several JSON format logs using a module called Zeek and configured the software Filebeat to classify and send these logs to our ElasticSearch cloud deployment. We then attempted to train a machine learning model on our cloud deployment, but were unsuccessful in doing so, as the deployment would not detect any anomalies in our traffic. Utilizing data views displaying the geographical origin of these logs, we were able to analyze and determine where the malicious traffic originated.

Project Overview

Milestones:

1. Determine roles of each member and create status report
2. Conduct research about different types of IDS
3. Decide on which IDS to deploy to container
4. Use IDS and Filebeat to send logs to Elastic
5. Train Machine Learning Model to recognize malicious traffic

Materials:

1. Elastic Deployment
2. Docker Container - Ubuntu
3. Malicious Traffic PCAP files
4. Normal Traffic PCAP files

Deliverables:

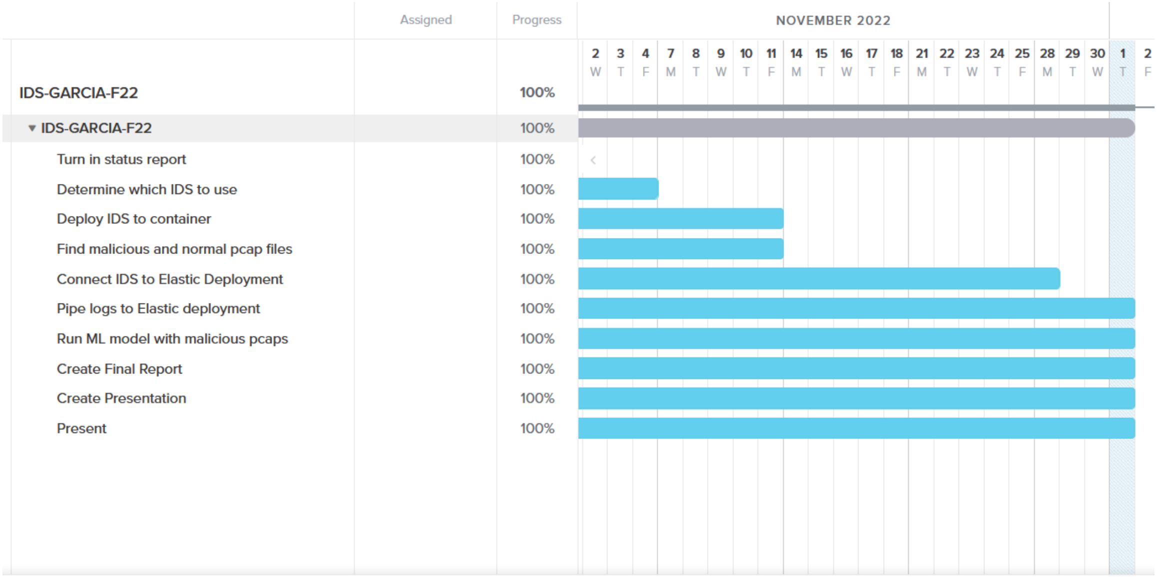
1. Status Report
2. Finalized Report
3. Presentation

Professional Accomplishments:

1. Learned how to deploy Zeek onto an Ubuntu Linux system.
2. Observed how machine learning can be used to predict anomalies.
3. Developed an understanding of how to utilize ElasticSearch for data manipulation.

Project Schedule Management

Gantt Chart:



	Assigned	Progress
IDS-GARCIA-F22		100%
▾ IDS-GARCIA-F22		100%
Turn in status report		100%
Determine which IDS to use		100%
Deploy IDS to container		100%
Find malicious and normal pcap files		100%
Connect IDS to Elastic Deployment		100%
Pipe logs to Elastic deployment		100%
Run ML model with malicious pcaps		100%
Create Final Report		100%
Create Presentation		100%
Present		100%

Management Board:

https://trello.com/invite/idsgarciaf22/ATTIb2cbc05befc1f9a590d0cec8ff96094c56FE29AB

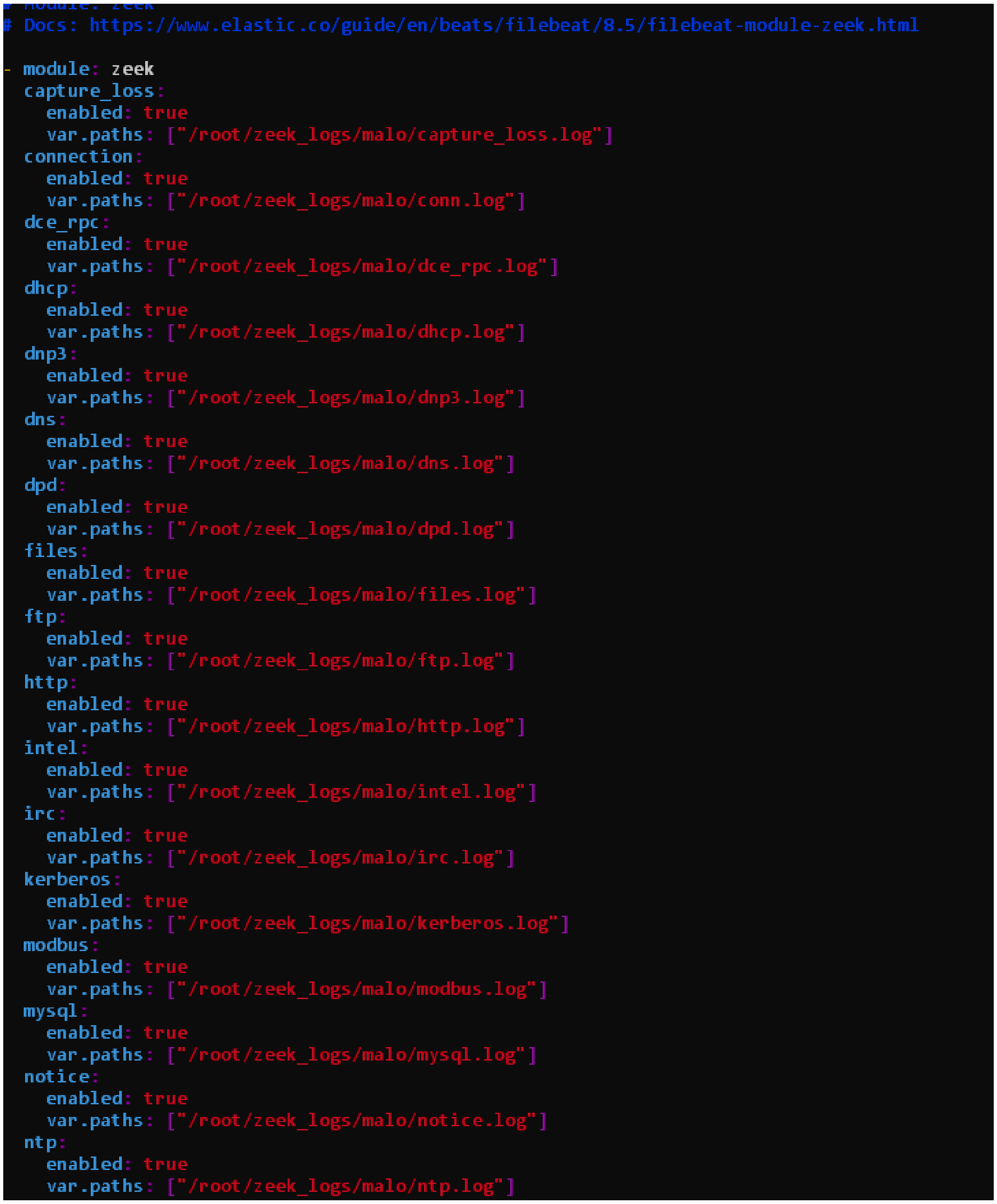
GitHub Repository:

https://github.com/dsgarza/IDS-GARCIA-F22

Software Overview

Zeek

After downloading the pcap files, we used Zeek to parse it into several log files and place them into a specified directory. We specified to Zeek to convert these log files to JSON format so that they would be readable to ElasticSearch after being sent by Filebeat.



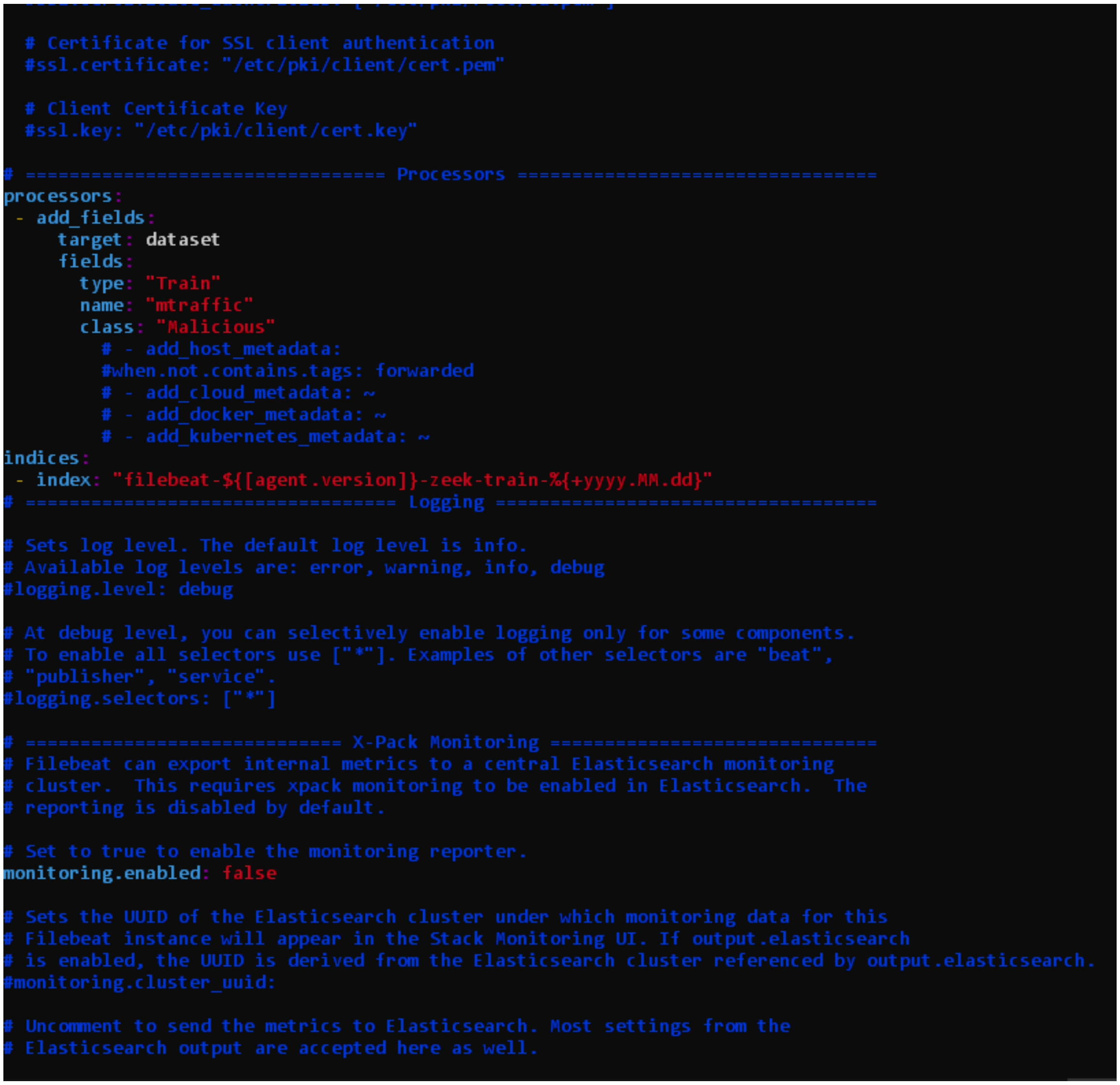
```
# Module: zeek
# Docs: https://www.elastic.co/guide/en/beats/filebeat/8.5/filebeat-module-zeek.html

- module: zeek
  capture_loss:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/capture_loss.log"]
  connection:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/conn.log"]
  dce_rpc:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/dce_rpc.log"]
  dhcp:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/dhcp.log"]
  dnp3:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/dnp3.log"]
  dns:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/dns.log"]
  dpd:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/dpd.log"]
  files:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/files.log"]
  ftp:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/ftp.log"]
  http:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/http.log"]
  intel:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/intel.log"]
  irc:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/irc.log"]
  kerberos:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/kerberos.log"]
  modbus:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/modbus.log"]
  mysql:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/mysql.log"]
  notice:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/notice.log"]
  ntp:
    enabled: true
    var.paths: ["/root/zeek_logs/malo/ntp.log"]
```

ElasticSearch

We utilized ElasticSearch to process our Zeek logs and display the traffic in different data views, as will be seen later in this report. ElasticSearch already has built-in tools to parse JSON logs from Zeek, so we did not need to make a pipeline to push the logs.

Filebeat

We configured Filebeat to both classify and send the logs that we were sending through it. As seen in the screenshot below, the first pcap we sent through it was classified as malicious and was meant to be used as training material for our machine learning deployment. The indices index option did not function properly and the logs always ended up under the default “filebeat-*” naming scheme.

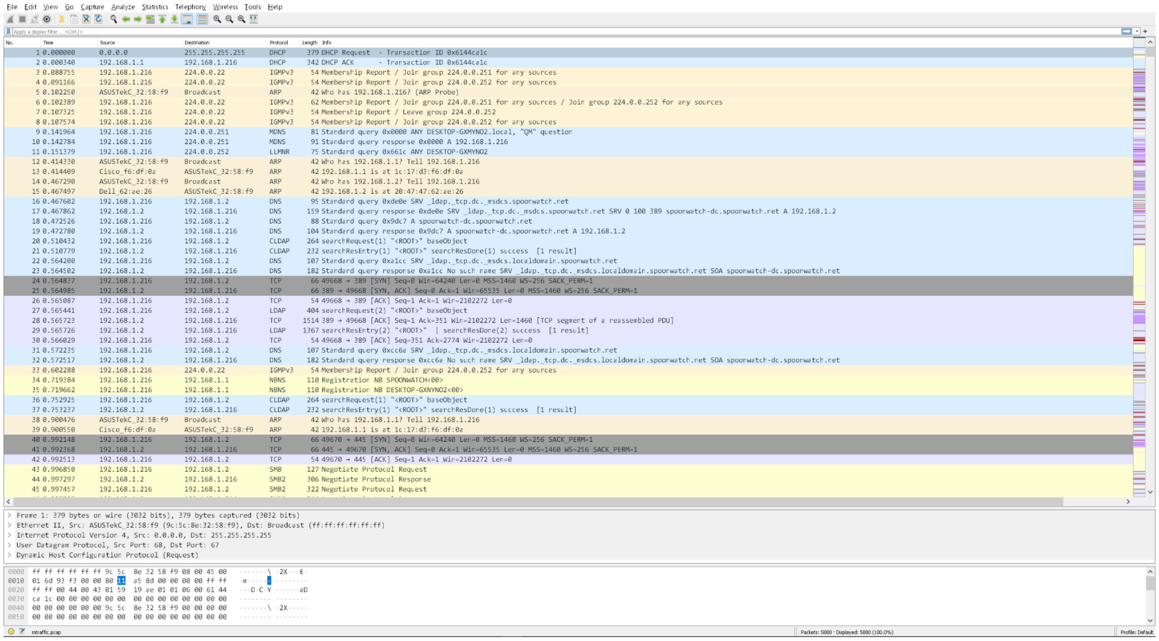


```
  # Certificate for SSL client authentication
  #ssl.certificate: "/etc/pki/client/cert.pem"

  # Client Certificate Key
  #ssl.key: "/etc/pki/client/cert.key"

# ================================= Processors =================================
processors:
 - add_fields:
     target: dataset
     fields:
       type: "Train"
       name: "mtraffic"
       class: "Malicious"
         # - add_host_metadata:
         #when.not.contains.tags: forwarded
         # - add_cloud_metadata: ~
         # - add_docker_metadata: ~
         # - add_kubernetes_metadata: ~
indices:
 - index: "filebeat-${[agent.version]}-zeek-train-%{+yyyy.MM.dd}"
# ================================== Logging ===================================

# Sets log level. The default log level is info.
# Available log levels are: error, warning, info, debug
#logging.level: debug

# At debug level, you can selectively enable logging only for some components.
# To enable all selectors use ["*"]. Examples of other selectors are "beat",
# "publisher", "service".
#logging.selectors: ["*"]

# ============================= X-Pack Monitoring ==============================
# Filebeat can export internal metrics to a central Elasticsearch monitoring
# cluster.  This requires xpack monitoring to be enabled in Elasticsearch.  The
# reporting is disabled by default.

# Set to true to enable the monitoring reporter.
monitoring.enabled: false

# Sets the UUID of the Elasticsearch cluster under which monitoring data for this
# Filebeat instance will appear in the Stack Monitoring UI. If output.elasticsearch
# is enabled, the UUID is derived from the Elasticsearch cluster referenced by output.elasticsearch.
#monitoring.cluster_uuid:

# Uncomment to send the metrics to Elasticsearch. Most settings from the
# Elasticsearch output are accepted here as well.
```

Elastic Analysis

PCAP Files

The pcap file that we used in this analysis was pushed to Elastic with known malicious traffic in order to possibly detect it apart from the normal traffic. We downloaded this pcap, SpoonWatch, from https://www.malware-traffic-analysis.net/.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
mtraffic.pcap
Packets: 5880 · Displayed: 5880 (100.0%)
Profile: Default

```
No.  Time      Source             Destination        Protocol Length Info
   1 0.000000  0.0.0.0            255.255.255.255    DHCP      379 DHCP Request  - Transaction ID 0x6144ca1c
   2 0.000340  192.168.1.1        192.168.1.216      DHCP      342 DHCP ACK      - Transaction ID 0x6144ca1c
   3 0.088755  192.168.1.216      224.0.0.22         IGMPv3     54 Membership Report / Join group 224.0.0.251 for any sources
   4 0.091166  192.168.1.216      224.0.0.22         IGMPv3     54 Membership Report / Join group 224.0.0.252 for any sources
   5 0.102250  ASUSTekC_32:58:f9  Broadcast          ARP        42 Who has 192.168.1.216? (ARP Probe)
   6 0.102389  192.168.1.216      224.0.0.22         IGMPv3     62 Membership Report / Join group 224.0.0.251 for any sources / Join group 224.0.0.252 for any sources
   7 0.107325  192.168.1.216      224.0.0.22         IGMPv3     54 Membership Report / Leave group 224.0.0.252
   8 0.107574  192.168.1.216      224.0.0.22         IGMPv3     54 Membership Report / Join group 224.0.0.252 for any sources
   9 0.141964  192.168.1.216      224.0.0.251        MDNS       81 Standard query 0x0000 ANY DESKTOP-GXMYNO2.local, "QM" question
  10 0.142784  192.168.1.216      224.0.0.251        MDNS       91 Standard query response 0x0000 A 192.168.1.216
  11 0.151379  192.168.1.216      224.0.0.252        LLMNR      75 Standard query 0x661c ANY DESKTOP-GXMYNO2
  12 0.414330  ASUSTekC_32:58:f9  Broadcast          ARP        42 Who has 192.168.1.1? Tell 192.168.1.216
  13 0.414409  Cisco_f6:df:0a     ASUSTekC_32:58:f9  ARP        42 192.168.1.1 is at 1c:17:d3:f6:df:0a
  14 0.467290  ASUSTekC_32:58:f9  Broadcast          ARP        42 Who has 192.168.1.2? Tell 192.168.1.216
  15 0.467497  Dell_62:ae:26      ASUSTekC_32:58:f9  ARP        42 192.168.1.2 is at 20:47:47:62:ae:26
  16 0.467602  192.168.1.216      192.168.1.2        DNS        95 Standard query 0xde0e SRV _ldap._tcp.dc._msdcs.spoonwatch.net
  17 0.467862  192.168.1.2        192.168.1.216      DNS       159 Standard query response 0xde0e SRV _ldap._tcp.dc._msdcs.spoonwatch.net SRV 0 100 389 spoonwatch-dc.spoonwatch.net A 192.168.1.2
  18 0.472526  192.168.1.216      192.168.1.2        DNS        88 Standard query 0x9dc7 A spoonwatch-dc.spoonwatch.net
  19 0.472780  192.168.1.2        192.168.1.216      DNS       104 Standard query response 0x9dc7 A spoonwatch-dc.spoonwatch.net A 192.168.1.2
  20 0.510432  192.168.1.216      192.168.1.2        CLDAP     264 searchRequest(1) "<ROOT>" baseObject
  21 0.510779  192.168.1.2        192.168.1.216      CLDAP     232 searchResEntry(1) "<ROOT>" searchResDone(1) success  [1 result]
  22 0.564200  192.168.1.216      192.168.1.2        DNS       107 Standard query 0xa1cc SRV _ldap._tcp.dc._msdcs.localdomain.spoonwatch.net
  23 0.564502  192.168.1.2        192.168.1.216      DNS       182 Standard query response 0xa1cc No such name SRV _ldap._tcp.dc._msdcs.localdomain.spoonwatch.net SOA spoonwatch-dc.spoonwatch.net
  24 0.564837  192.168.1.216      192.168.1.2        TCP        66 49668 → 389 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
  25 0.564985  192.168.1.2        192.168.1.216      TCP        66 389 → 49668 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
  26 0.565087  192.168.1.216      192.168.1.2        TCP        54 49668 → 389 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
  27 0.565441  192.168.1.216      192.168.1.2        LDAP      404 searchRequest(2) "<ROOT>" baseObject
  28 0.565723  192.168.1.2        192.168.1.216      TCP      1514 389 → 49668 [ACK] Seq=1 Ack=351 Win=2102272 Len=1460 [TCP segment of a reassembled PDU]
  29 0.565726  192.168.1.2        192.168.1.216      LDAP     1367 searchResEntry(2) "<ROOT>"  | searchResDone(2) success  [1 result]
  30 0.566029  192.168.1.216      192.168.1.2        TCP        54 49668 → 389 [ACK] Seq=351 Ack=2774 Win=2102272 Len=0
  31 0.572235  192.168.1.216      192.168.1.2        DNS       107 Standard query 0xcc6a SRV _ldap._tcp.dc._msdcs.localdomain.spoonwatch.net
  32 0.572517  192.168.1.2        192.168.1.216      DNS       182 Standard query response 0xcc6a No such name SRV _ldap._tcp.dc._msdcs.localdomain.spoonwatch.net SOA spoonwatch-dc.spoonwatch.net
  33 0.602288  192.168.1.216      224.0.0.22         IGMPv3     54 Membership Report / Join group 224.0.0.252 for any sources
  34 0.719384  192.168.1.216      192.168.1.1        NBNS      110 Registration NB SPOONWATCH<00>
  35 0.719662  192.168.1.216      192.168.1.1        NBNS      110 Registration NB DESKTOP-GXNYNO2<00>
  36 0.752925  192.168.1.216      192.168.1.2        CLDAP     264 searchRequest(1) "<ROOT>" baseObject
  37 0.753237  192.168.1.2        192.168.1.216      CLDAP     232 searchResEntry(1) "<ROOT>" searchResDone(1) success  [1 result]
  38 0.900476  ASUSTekC_32:58:f9  Broadcast          ARP        42 Who has 192.168.1.1? Tell 192.168.1.216
  39 0.900550  Cisco_f6:df:0a     ASUSTekC_32:58:f9  ARP        42 192.168.1.1 is at 1c:17:d3:f6:df:0a
  40 0.992148  192.168.1.216      192.168.1.2        TCP        66 49670 → 445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
  41 0.992368  192.168.1.2        192.168.1.216      TCP        66 445 → 49670 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
  42 0.992513  192.168.1.216      192.168.1.2        TCP        54 49670 → 445 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
  43 0.996850  192.168.1.216      192.168.1.2        SMB       127 Negotiate Protocol Request
  44 0.997297  192.168.1.2        192.168.1.216      SMB2      306 Negotiate Protocol Response
  45 0.997457  192.168.1.216      192.168.1.2        SMB2      322 Negotiate Protocol Request
```

```
> Frame 1: 379 bytes on wire (3032 bits), 379 bytes captured (3032 bits)
> Ethernet II, Src: ASUSTekC_32:58:f9 (9c:5c:8e:32:58:f9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
> Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
> User Datagram Protocol, Src Port: 68, Dst Port: 67
> Dynamic Host Configuration Protocol (Request)
```

```
0000  ff ff ff ff ff ff 9c 5c  8e 32 58 f9 08 00 45 00   ·······\ ·2X···E·
0010  01 6d 93 f3 00 00 80 11  a5 8d 00 00 00 00 ff ff   ·m······ ········
0020  ff ff 00 44 00 43 01 59  19 ae 01 01 06 00 61 44   ···D·C·Y ······aD
0030  ca 1c 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
0040  00 00 00 00 00 00 9c 5c  8e 32 58 f9 00 00 00 00   ·······\ ·2X····
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
```

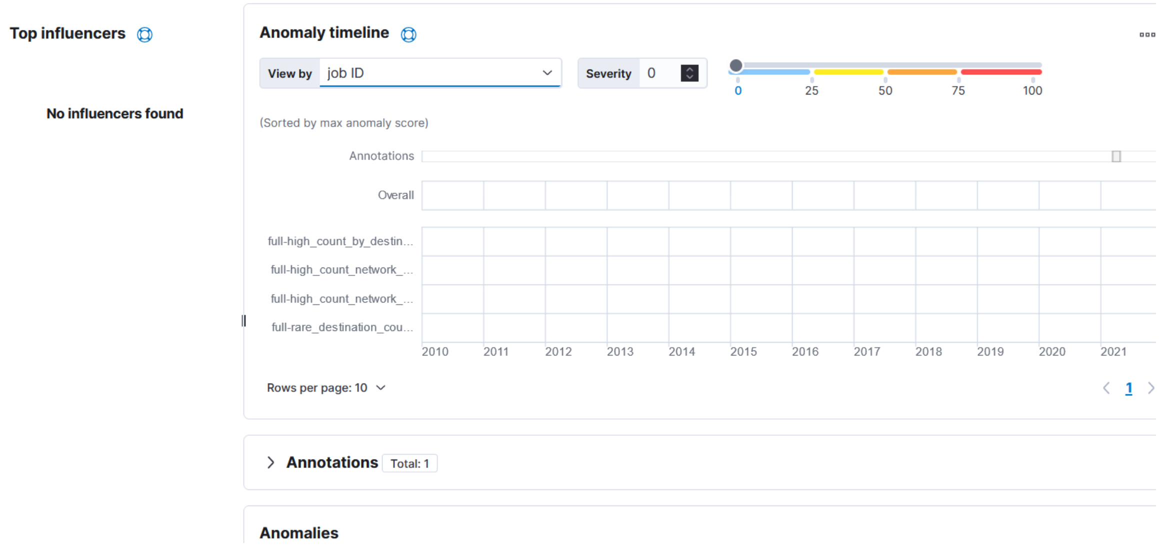
Difficulties

We had several issues getting Filebeat to upload Zeek logs. At first, we believed the issue to be an error in our configuration files. Because of this false notion, we spent a large amount of time recreating our containers. This included completely rewriting our Filebeat configuration file and Zeek configuration file to ensure there were no issues. After several failed attempts, we worked with Dr. Parra to determine that our deployment had an older version of Stack Management that was incompatible with the newer Filebeat install.

We also had issues getting our machine learning model to work. We created an Anomaly Detection job with a malicious pcap. However, the job detected no anomalies. We were instructed by Dr. Parra that the model needed to be trained before it could detect any anomalies.

Machine Learning

As seen in the screenshot below, there were no anomalies detected by ElasticSearch despite the pcap file being used having known malicious traffic. We ran normal traffic through this process first with expected results, but were not able to solve this issue.



Top influencers

No influencers found

Anomaly timeline

View by job ID

Severity 0

(Sorted by max anomaly score)

Rows per page: 10

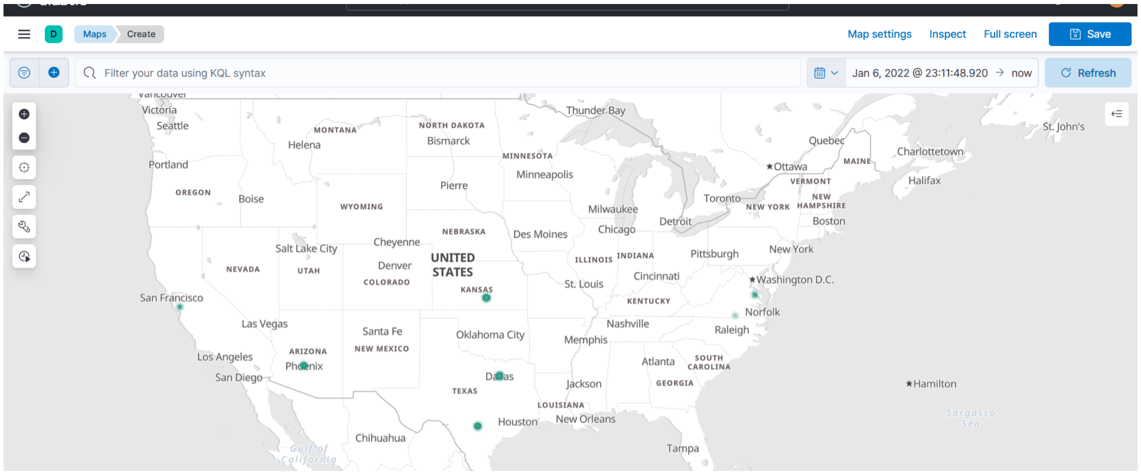
Annotations Total: 1

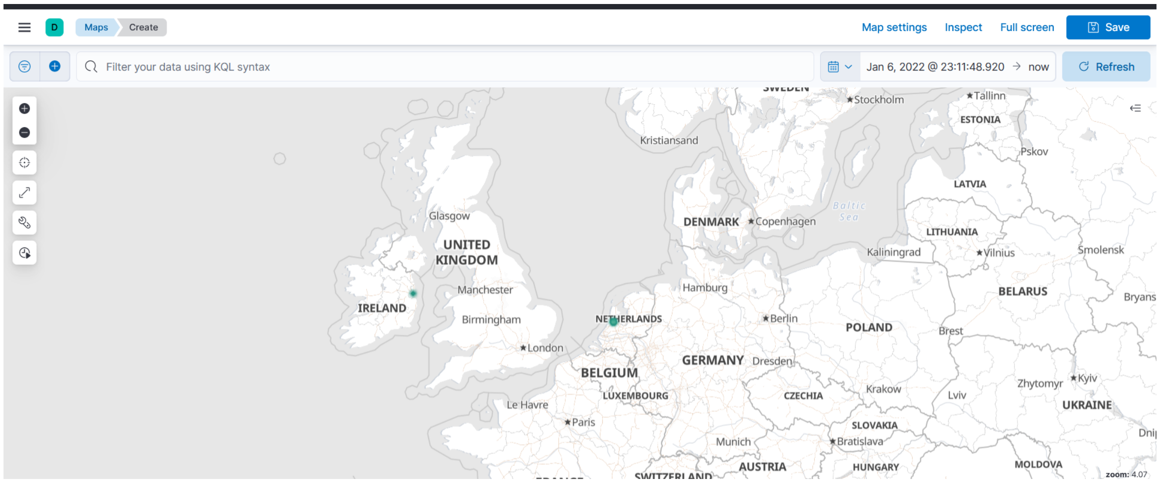
Anomalies

0
25
50
75
100
Annotations
Overall
full-high_count_by_destin...
full-high_count_network_...
full-high_count_network_...
full-rare_destination_cou...
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
1

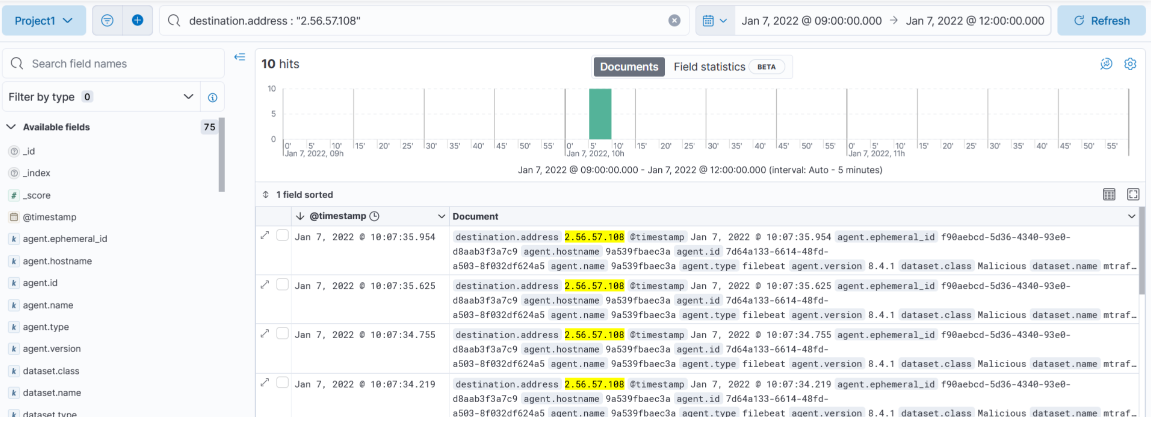
Data Views

Outside of the continental United States, the only traffic in the SpoonWatch pcap comes from Ireland and the Netherlands. What stood out about this traffic was that it was sent in small increments while other hot spots in the United States sent hundreds of packets. This anomaly intrigued us and caused us to investigate further.





Here, in the Discover tab, we learned that the destination IP from the Netherlands is 2.56.57.108.



destination.address : "2.56.57.108"

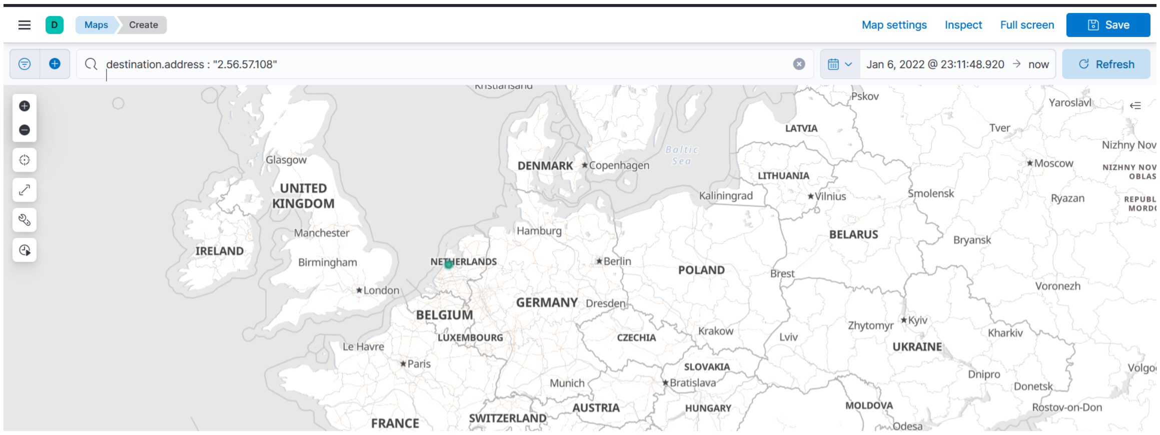
Jan 7, 2022 @ 09:00:00.000 → Jan 7, 2022 @ 12:00:00.000

10 hits

Jan 7, 2022 @ 09:00:00.000 - Jan 7, 2022 @ 12:00:00.000 (interval: Auto - 5 minutes)

1 field sorted

```
@timestamp                  Document
Jan 7, 2022 @ 10:07:35.954  destination.address 2.56.57.108 @timestamp Jan 7, 2022 @ 10:07:35.954 agent.ephemeral_id f90aebcd-5d36-4340-93e0-
                            d8aab3f3a7c9 agent.hostname 9a539fbaec3a agent.id 7d64a133-6614-48fd-
                            a503-8f032df624a5 agent.name 9a539fbaec3a agent.type filebeat agent.version 8.4.1 dataset.class Malicious dataset.name mtraf…
Jan 7, 2022 @ 10:07:35.625  destination.address 2.56.57.108 @timestamp Jan 7, 2022 @ 10:07:35.625 agent.ephemeral_id f90aebcd-5d36-4340-93e0-
                            d8aab3f3a7c9 agent.hostname 9a539fbaec3a agent.id 7d64a133-6614-48fd-
                            a503-8f032df624a5 agent.name 9a539fbaec3a agent.type filebeat agent.version 8.4.1 dataset.class Malicious dataset.name mtraf…
Jan 7, 2022 @ 10:07:34.755  destination.address 2.56.57.108 @timestamp Jan 7, 2022 @ 10:07:34.755 agent.ephemeral_id f90aebcd-5d36-4340-93e0-
                            d8aab3f3a7c9 agent.hostname 9a539fbaec3a agent.id 7d64a133-6614-48fd-
                            a503-8f032df624a5 agent.name 9a539fbaec3a agent.type filebeat agent.version 8.4.1 dataset.class Malicious dataset.name mtraf…
Jan 7, 2022 @ 10:07:34.219  destination.address 2.56.57.108 @timestamp Jan 7, 2022 @ 10:07:34.219 agent.ephemeral_id f90aebcd-5d36-4340-93e0-
                            d8aab3f3a7c9 agent.hostname 9a539fbaec3a agent.id 7d64a133-6614-48fd-
                            a503-8f032df624a5 agent.name 9a539fbaec3a agent.type filebeat agent.version 8.4.1 dataset.class Malicious dataset.name mtraf…
```



destination.address : "2.56.57.108"

Jan 6, 2022 @ 23:11:48.920 → now

It was at this point that the limitations of only using data views began to become more clear. A trained machine learning model could pick out the malicious traffic. We resigned to checking the answer key file to determine the actual source of the malware. The answer key revealed that the threat actor was using a malware called OskiStealer to steal the host's information.

Overview of Findings

Potential Applications

Due to its low cost, at least relative to other NIDS, a combination of Zeek and Elastic could be a valid option for home networks or even some small businesses. However, as the business grows, so would one's need to look into more robust security solutions, such as Crowdstrike or Arctic Wolf.

Conclusion

Zeek is a great option for processing logs, and was especially useful in producing them in JSON format. While the configuration was arguably time consuming, it proved to be very responsive during use. With functionality such as live processing of network traffic, to our use of processing offline logs, it is very versatile and rich in features.

ElasticSearch is a powerful tool, not only in the world of data analytics but also in the world of security. Even though we were unable to get the machine learning model working, we were almost able to pinpoint exactly where the malicious traffic originated. If a team was more successful in implementing a trained machine learning model, they would be able to automate detecting anomalies in live network traffic and configure Kibana to send alerts back to them.

REFERENCES

Eric. “Zeekurity Zen – Part VIII: How to Send Zeek Logs to Elastic.” *Ericooi.com*, 6 Sept. 2022, https://www.ericooi.com/zeekurity-zen-part-viii-how-to-send-zeek-logs-to-elastic/.

“Traffic-Analysis.net.” *Malware*, https://www.malware-traffic-analysis.net/.

Young, Michael. “Collecting and Analyzing Zeek Data with Elastic Security.” *Elastic Blog*, Elastic, 29 July 2021, https://www.elastic.co/blog/collecting-and-analyzing-zeek-data-with-elastic-security.