Elastic Cloud Anomaly Detection

Testing on APT 29 and Feasibility Recommendations

ANOMALOUS

Abstract

Analyzing network traffic is an integral part of ensuring the security of a network. However, traditional signature-based detection does not protect against novel attacks. Through the use of an Elastic Stack deployment in Elastic Cloud (Software as a Service) composed of Elasticsearch, Logstash, and Kibana, we can use their machine learning nodes to detect anomalous traffic as it traverses a network. Additionally, features in the Elastic stack deployment allow us to generate alerts and supply recommended follow-up actions based on the severity and type of alert. However, Elastic is not a perfect one size fits all tool. We initially found that Elastic would only succeed at machine learning anomaly detection if the team setting up the detection had extensive knowledge of what they were looking for. After extensive research and trial/error, we did find that some success could be achieved with methods that require fewer resources and background knowledge (such as queries). Therefore, we determined that Elastic anomaly detection is a decent middle ground tool to bridge the gap between manual searches and queries, however, it needs certain data characteristics to meet its full potential.

Introduction

This project was sponsored by the Cybersecurity and Infrastructure Security Agency (CISA), within the Department of Homeland Security (DHS). The overarching goal was to create an anomaly detection system as a proof of concept to help CISA in the ongoing development of their own Elastic based Anomaly Detection system. CISA supports federal agencies by providing various network security services along with an operational dashboard to host results from these services.

This research has identified various pitfalls and dead ends that CISA has yet to encounter. To fulfill this goal, we attempted to find or generate our own malicious network activity to use in Elastic. Generating datasets was not successful, but after a lengthy search and with assistance from CISA we found the Mordor Project [1], created by Roberto Rodriguez (@Cyb3rWard0g) and Jose Luis Rodriguez (@Cyb3rPandaH). The Mordor Project provides datasets of security events generated from simulated attack techniques. These crucially include additional context in the form of both contextual writeups and non-malicious events that occur around it. The name is based off Mordor from the Lord of the Rings, the place where the evil forces of Sauron gathered [1]. Within the Mordor Project we used some small datasets as practice and proof of concept. Then we selected one of the large datasets about APT 29 to be the dataset we use to build our final product.

APT 29 / Cozy Bear

APT 29 is a threat group linked to the Russian Federation that is known to target political organizations and think tanks. Their most widely known attack was against the Democratic National Committee (DNC) during the 2016 US Presidential Elections [5].

CISA has an interest in understanding this attack group and the payloads they use, which makes this dataset ideal for us to work on. The Mordor Dataset on APT 29 was created to emulate adversarial attacks and techniques for 2019 ATT&CK Evaluations of multiple organizations in Nashua, NH (Day 1) and Scranton, PA (Day 2).

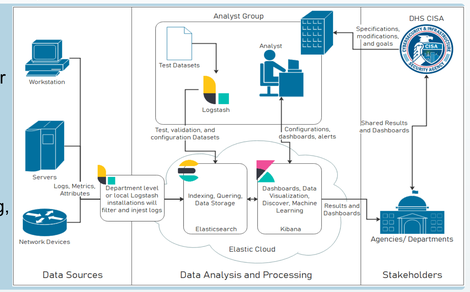
APT 29 payloads include CosmicDuke, MiniDuke, SeaDuke/SeaDaddy CozyDuke/CozyCar, and Hammertoss [4]. Duke is a set of malware that has been used in attacks since 2008 beginning in the Chechnya Informational Center Incident and has affected many high-level organizations. APT 29 is thought to have originated these tools [3].

Design

Our design uses Elastic Cloud as our main provider, with our most used products from Elastic being: Elasticsearch, Kibana, Logstash, Anomaly Detection Jobs and, Data Visualizer.

Concept of Operations

The concept for this project follows three main groups; Data Sources, Data Analysis and Processing, and Stakeholders



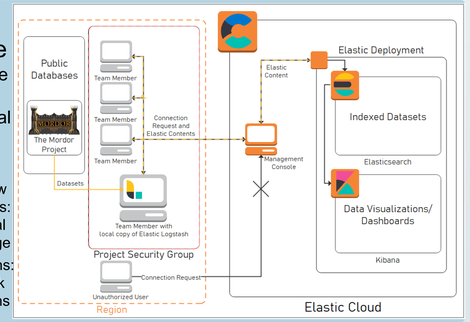
The data sources in a production environment would be any device the stakeholders want to monitor. These data sources will use Logstash to ingest log files into the appropriate Elastic Deployment. From there an Analyst group will monitor, maintain, and test the Elastic deployment using Elasticsearch as well as Kibana. The analysts will use Kibana to produce dashboards for both their use as well as the stakeholders. Who will also have access to the Elastic deployment. From there CISA will work with the other stakeholders to both create a government wide dashboard as well as refine the priorities of the operating analyst group.

Product Architecture

Our architecture consists of two groups; the local team and the Elastic Cloud.

Yellow Connections: Informational Exchange

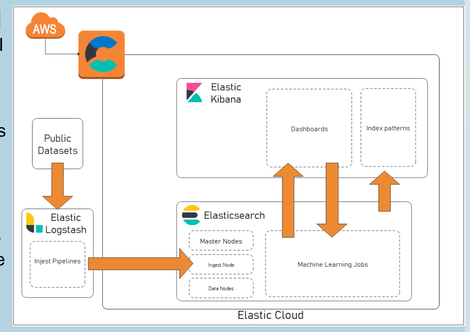
Black Connections: Network Connections



Our architecture is secured using Elastic Cloud user management, only approved members of the team will be issued credentials with various permissions. Members of the local team will be able to access the deployment and use Logstash to upload into the Elastic deployment. Within Elastic Cloud our architecture relies of Elasticsearch to ingest and manage our datasets, while Kibana is used for anomaly detection jobs, creating data visualizations, and creating dashboards

Data Model

Our data model is simple: Logstash is used to upload datasets or logs to our Elastic deployment where Elasticsearch and Kibana are used to analyze and display the results



Implementation

After initial failed attempts to find or generate appropriate datasets, CISA pointed us towards the Mordor Project [1], where we used 6 smaller datasets to test our deployments. We ran these datasets through anomaly detection features with minimal success due to the limited size and time span of the datasets, which makes it impossible for the detection system to establish a "normal" reading. So, we decided to identify one of the large datasets available through the Mordor Project.

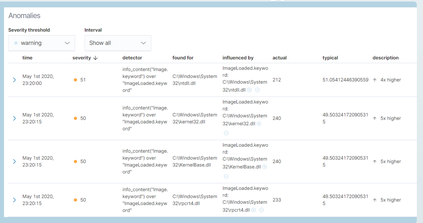
The dataset we selected was a simulated dataset developed after analyzing the threat groups historical attack pattern. The preattack on these datasets consisted of a spear-phishing campaign where the appropriate attack tool is selected based on the results [4].

Using the MITRE ATT&CK® Framework, the attack consists of [4]:

1. Login to victim system and launch PUPY server (Initial Access)
2. Compress user login information files and export through PUPY (Credential Access)
3. Allows for reverse HTTPS from a malicious web server to drop files into a victim machine (Execution)
4. Hide and prevent detection (Defense Evasion)
5. Gain access to credentials by stealing private keys (Credential Access)
6. Copy payload into a client (Lateral Movement)
7. Execute SeaDuke, a backdoor created by APT29 to continuously have access to the system (Persistence & Execution)

From this, we deduced that the anomalistic data would be in features that consist of attempts to log in, connection with their web server, processes to copy files to/from their server, and malicious services that relate to APT 29. We determined that anomalistic data was best identified from using the Categorization, Population, and Advanced machine learning jobs.

Example of anomalies found for count by "Image.keyword" with influencer "ImageLoaded"



Data Frame Analytics enable us to determine outliers for specific features and validate these results. However, this is only possible to confirm on a per-feature basis, requiring extensive manual review.

One of the biggest issues we had during the project was getting Logstash to work. Even though CISA had custom configuration files they shared, it would not work. This was due in part for two reasons, one that we needed to set it to read mode so that it would only try and ingest one file, and that we had to use the Elasticsearch endpoint. The most helpful asset in creating our config was the Logstash plugin repository [2]. A copy of our config files along with a step-by-step guide can be shared upon request.

Once we had finished setting up Logstash and Analyzing our data we moved on to creating a dashboard. Our dashboard was meant for a SOC team lead to give an overview of what kind of resources were needed to be assigned to a given log or dataset.

Dashboard of four visualizations consisting of: Destination IP Addresses, Destination Ports, Source Addresses, and Subject Username

Verification and Validation

We verified the integrity of the datasets by manually exploring the datasets using Elastic data visualizer. We validated our results looking for false positives initially through cross referencing Data Frame Analytics and our anomalous results. Some false-positives were recognized from this process, such as Microsoft Teams Installer which likely has no relation to APT 29. A more in-depth manual review was conducted to confirm our results and make recommendations accordingly.

Recommendations

Based on our work, we cannot recommend machine learning anomaly detection as a sole data-analysis product. Instead, machine learning should be used as an additional resource on the tool belt of a security team that wants more verification and validation for their manual work. This is due to the large amount of investigation and analysis required for anomaly detection to work. However, our perspective is limited to the Mordor Project datasets analyzed, some of which were limited in size. What we can say is that machine learning-based anomaly detection works best when using large datasets with a small amount of contextual understanding required. For example, identifying one type of denial-of-service attack requires recognizing an extreme increase in traffic, which requires a very limited contextual understanding. Alternatively, identifying an attack that involves utilizing specific functions in a specific order requires a high contextual understanding. Given that trained analysts are much better at understanding the context of an attack, the attack in the highly contextual example could be more easily identified using a human-made query.

Future Work

Future work that needs to be completed in this project mainly needs to address sample size. Due to the requirements and timeline of this project we could not examine a sufficient number of datasets. Another team using our setup could accomplish this easily by examining the 83 other small datasets available from the Mordor Project if they find a way to get around the ineffectiveness of small datasets. They could also spend the time working on the other large-scale dataset the Mordor Project has based on APT 3. Although we did not find a lot of success using single metric and multi-metric anomaly job features, future research can also be done on those features as they can be useful in other scenarios.

Acknowledgements

The team would like to thank the entire CISA team for their support, guidance as well as the opportunity to work on this project. We would like to thank the team at the Mordor Project for their work on their public datasets. Finally, we would like to thank our Capstone professor for his constant support and mentorship on this project.

References

1. R. Rodriguez and J. L. Rodriguez, "Introduction," Introduction - 2020. [Online]. Available: https://mordordatasets.com/introduction.html. [Accessed: 12 Apr-2021].
2. Elasticsearch B.V., "Logstash Introduction," Elastic. [Online]. Available: https://www.elastic.co/guide/en/logstash/current/introduction.html. [Accessed: 12 Apr-2021].
3. "The Dukes: 7 Years Of Russian Cyberespionage," F-Secure, 17-Sep-2015. [Online]. Available: https://blog-assets.f-secure.com/wp-content/uploads/2020/03/18122307/F-Secure_Dukes_Whitepaper.pdf. [Accessed: 13 Apr-2021].
4. Mitre-Attack, "mitre-attack/attack-arsenal/adversary_emulation/APT29/Emulation_Plan/Day1/," GitHub, 23-Jun-2020. [Online]. Available: https://github.com/mitre-attack/attack-arsenal/tree/master/adversary_emulation/APT29/Emulation_Plan/Day%201. [Accessed: 13 Apr-2021].
5. Kaspersky Lab. , "What's behind APT29?," Kaspersky MITRE ATT&CK [Online]. Available: https://www.kaspersky.com/enterprise-security/mitre/apt29. [Accessed: 13 Apr-2021].