Cybercriminal Network Building: An Automated Solution for Cyber Threat Analysis

CS4991 Capstone Report, 2024

Kevin Carlson
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
kmc6am@virginia.edu

ABSTRACT

Cybersecurity analysts often only have access to individual data nodes when responding to a detected threat. Connecting these nodes to potential threat actors is a labor-intensive process in a time-sensitive environment. My team and I developed an end-client application for generating meaningful connections between a queried data point and filtered datasets using a refined clustering algorithm. Our solution utilized an in-house, synthetically generated Personally Identifiable Information (PII) dataset to represent a reasonably practical dataset available to the client. We employed a density-based spatial clustering of applications with noise (DBSCAN) algorithm to automatically develop multi-level relationships between nodes, with the end product displaying these results in graphical interfaces. Given several test cases, our model correctly identified 89% of related data points when queried on a single PII attribute (e.g. blockchain address, MAC address, email). The project's next steps involve expanding the potential data types to be handled by the algorithm and further refining the recognition of relationships by exploring other means of clustering information.

1. INTRODUCTION

In 2023 alone, an estimated 353 million people were impacted by online data breaches [1]. The increasing number of individuals interacting with the Internet of Things (IoT) daily lays the foundation for a significant amount of personal information to be shared across the Internet. As technology expands, innovations are released, allowing new possible attack vectors to be exploited to gain access to user information.

Take, for example, the Capital One data breach in 2019. Hackers exfiltrated files containing approximately 100,000 social security numbers, and 80,000 data points of PII from customers and credit card applications such as addresses, credit transactions, account balances, and more [2]. Once accessed, individuals or organized hacking groups can leverage this information very quickly to perform a malicious action and cover their tracks before a mitigation effort can even be established. Suppose such a group can access sensitive information, such as financial information or intellectual property. In that case, they can either sell this information on the dark web or extort the owner of the information to pay a ransom to obtain the stolen data.

In 2021, a cybercriminal group called DarkSide hacked its way into the secure servers of Colonial Pipeline through an exposed VPN password. The group exfiltrated 100Gb of data in two hours and proceeded to inject ransomware into the Colonial Pipeline's IT network, freezing server access until a ransom was paid [3]. Not only did Colonial Pipeline decide to pay the

ransom of 75 bitcoins (USD 4.4 million), but due to the importance of the pipeline for many industries in the US, fuel shortages in the airline industry, and gas price increases for consumers along the East Coast were seen instantly [3]. These instances of cyber attacks prove the insistent need for an automated cyber threat analysis tool.

2. RELATED WORKS

My team and I researched the current market for threat-hunting solutions and found multiple applications and websites serving similar solutions, but varied enough to allow our project to fulfill the specific niche demanded by the client. One such popular website, “haveibeenpwned”, provides a more consumer-friendly and bare-bones means of querying PII against their databases to determine whether the target information has been a part of any data breaches [4]. Since it is targeted towards the general public, this website primarily only searches on discrete personal data points such as emails or phone numbers, which are unlikely to be exposed in a threat-hunting operation.

Another researched alternative is Constella Intelligence. The firm offers several solutions, including deep OSINT (open-source intelligence) investigations on bad actors and insiders using its proprietary identity dataset [5]. This firm was currently being utilized for similar projects by the company I was employed by during this project. After speaking with sources working with Constella, we determined they did not provide a solution that offered visualization generation or customizable scalability for the breadth of the search algorithm, both of which were major aspects of our project.

3. PROJECT DESIGN

When a sensitive data point is discovered and flagged as compromised, a team of analysts must work quickly to backtrace the source and identify the malicious hackers. Typically only having a singular “breadcrumb” of data to start with (e.g. IP address, MAC address, cryptocurrency ID), filtering through existing and live data streams to identify connected pieces of information can be intensive and time-consuming, in an environment where time cannot be wasted. When these malicious hackers are tied to cybercriminal organizations, important connections can be made between data points. All of this information can be slowly pieced together by analysts by working through data, manually creating a web-like structure of connections between cybercriminal data points as information is discovered. The project design pipelines this process into an automated function by allowing the analyst to input a starting data point and receive an interactive web-like graph showing connections among relevant data points to the target.

Our team’s project was concetenated into four distinct steps in an agile development workflow: Data synthesis, automated processing of data, spatial clustering of relevant data points, and interactive display of results. Since the data to be clustered is classified as PII, it was impractical to use an existing dataset due to privacy violations. Our team used an open-source, artificial intelligence data generation tool called Mockaroo [6] to synthesize approximately 250,000 lines of simulated breach data using custom scripts for the greatest level of reasonability. For example, data on cryptocurrency blockchain transactions would include two Bitcoin wallet addresses and an amount, in cryptocurrency, and data from a simulated Capital One breach included account, credit card, and social security numbers. We also fabricated 20 individual test personas with a random number of relevant data attributes and injected them into the dataset. These attributes were then used for testing our final design for precision and recall. The second

step of the project was to process the finalized dataset and prepare it to be fed into our machine-learning model. Processing the data consisted of typecasting variables as strings or floats, splitting text field attributes into list elements, and removing trivial characters. For example, the email address "john_doe4@gmail.com" would be processed into ['john', 'doe', '4', 'gmail.com']. Additionally, comparable fields such as usernames and emails, where a threat actor might share an overlap of identifying information, were copied into separate lists for joint encoding.

Word2vec, a popular text-embedding technique that represents individual words as dense vectors in a continuous vector space, encoded all string-type variables. One-hot encoding was used for all fields needing an exact match to be considered related (e.g. public IP address 46.62.56.84 should not be deduced as related to 46.62.56.90). Once all columns were encoded, we fed the dataset into our clustering algorithm. Our team utilized DBSCAN (Density-Based Spatial Clustering of Applications with Noise), a popular clustering algorithm in machine learning for grouping data points based on their distance in a dimensional space. This algorithm was run on one or two data variables at a time to reduce the dimensionality, and thus runtime, of the program. Each execution of the algorithm yielded a separate file with the encoded data variable(s), and a cluster identifier for each line in the dataset, which was gathered from the algorithm based on what data points were grouped together.

The DBSCAN algorithm has two main hyperparameters: epsilon (the minimum Euclidean distance two points should be to be considered in the same cluster) and minPoints (the minimum number of data points needed to form a dense region and consider clustering). Both variables were tested extensively with different values to refine the recall of the algorithm. Once all separate runs of the algorithm were complete, the program used the line numbers of the data points to transitively cluster relevant information together. For example, line 15,000 with data [168.42.58.24, ['stealth', 'penguin'] in the IP address, username fields respectively, clustered with line 16,000 and data ['thispenguinisstealthy', 99245652] in the username, bitcoin wallet address fields respectively. In another iteration of DBSCAN, line 16,000 clustered with line 17,000 with data [99245652, 99836712] in two Bitcoin wallet address fields. This step in our program would assign a global cluster ID to all lines (15,000, 16,000, and 17,000). This was a crucial step in developing the resultant network necessary for satisfying the problem statement.

While refinement of the DBSCAN algorithm on our dataset continued, our team developed an interactive web application for the end user. In the application, the user was able to identify a specific data attribute to search for, input the data attribute, and receive both a tabular and graphical view of the relevant connections to the queried data point. The tabular view simply displays the clustered data as a spreadsheet, and the end user can download this as a .xlsx file. The graphical representation used Pyvis, a Python library for visualizing network connections. Each node in the graph represented a line of data in the dataset, while each edge between nodes represented that the two nodes were clustered together in an iteration of DBSCAN.

4. RESULTS

In our specific client's use case, the breadth of information was deemed significantly more important than the precision of such data. Therefore, we used a recall formula for assessing the quantitative results our program produced. After extensive tailoring of our algorithm to increase recall, we found that our

program could correctly recall 89% of clustered information relating to the 20 injected personas. Further qualitative analysis revealed that this information was also correctly displayed through the graphical view in the web application, with all relevant information found through the algorithm being displayed appropriately. We found these results to be sufficient to satisfy the problem set.

5. CONCLUSION

This project proved successful in meeting the needs of the company outlined in the problem statement. This was accomplished through the four steps data synthesis, data processing, spatial clustering, and display. While the project design described the unique methods used for each of these steps, different techniques could be used to compare recall. The technical knowledge I gained through the course of this project was invaluable for future work. Additionally, the corporate skills and resources I utilized made me a better teammate, coworker, and employee for future assignments.

Consultation with members of a specialized security team throughout the project timeline helped us tailor the solution to become a field-ready application. At the end of the project, these team members verified the end product was a relevant tool that had immediate use for their team. Access to the solution my team and I created directly helping the mission of the company's top security team was a uniquely gratifying perspective to find.

6. FUTURE WORK

Further research could allude to deeper testing on hyperparameter valuation for the DBSCAN algorithm. It would also be useful for my team and I to conduct more research on algorithms to use instead of DBSCAN to perform a comparative analysis on performance, scalability, modularity, and level of recall. Should our product be brought to production with the design as is, it would be useful to make the product fully autonomous. This would mean allowing the client to import their own dataset, and/or hosting a database to collect live data streams that can run on our product in real-time. Another step would be to introduce an AI-powered automated processor to handle new types of data and encode them accordingly for insertion into the clustering algorithm. These steps would all make the product easier and more powerful for the client.

REFERENCES

[1] Petrosyan, A. (2024, February 12). *Number of data breaches and victims U.S. 2023*. Statista. https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/

[2] Llamas, M. (2023, April 17). *Biggest data breaches in history: Top 6 breaches in U.S.* Consumer Notice, LLP. https://www.consumernotice.org/data-protection/breaches/biggest-in-history/

[3] Kerner, S. M. (2022, April 26). *Colonial pipeline hack explained: Everything you need to know*. WhatIs. https://www.techtarget.com/whatis/feature/Colonial-Pipeline-hack-explained-Everything-you-need-to-know

[4] Have I Been Pwned? (haveibeenpwned.com)

[5] Constella (constella.ai)

[6] Mockaroo (mockaroo.com)