Data Metrics Dashboard Security Analysis

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Davin Um
Fall, 2020

Technical Project Team Members
Davin Um

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Signature _____Davin Um_____ Date __3/26/21__

Approved _____Cohoon, James_____ Date __12/7/20_____
Department of Computer Science

ABSTRACT

As the threat of cyber-attacks increase with the changes that happened due to COVID, there needs to be an effective way to communicate between the cybersecurity company and the client without requiring complex explanation. The research will provide better understanding for the user to specify which threats are happening to their current system and the company directly analyzes the problem through metrics dashboard, which will provide information related to network traffic, malware, fraud, etc. Previous research and projects have mainly focused on the company itself solving the problem when threats happened and not alerting the user what specific threats were detected. With this research, the user is able to understand which threats happened and take actions that can prevent the attack. The COVID pandemic has caused employees to work remotely, which introduced cyber threats to be more approachable to devices without secure internet connection or poor security systems. With the experiences related to courses such as Intro to Cybersecurity and HCI in Software Development, I will incorporate the basic knowledge of cybersecurity and user experience to create a data metrics panel that will analyze the threat data. Using basic knowledge I've learned from HCI class, I will conduct user research to find out what the real users desire, build into a design process such as making wireframes and prototypes, and conduct user tests to receive feedback. The research will involve pre-existing cyber security data-sets that are available to the public. With the given data, it will be transported to a specific database such as InfluxDB, and the database will be utilized to create visualizations of the data, showing log line numbers or number of occurrences for different kinds of threats.

1 Introduction

The COVID-19 has impacted businesses critically, which have caused employees to work in remote environments. While these workers are working separately, the hackers have been trying to take advantage of the situation where the workers are ill prepared for security. According to Furnell and Shah [1], the data indicates that 30% of UK companies are well prepared for user education and awareness. This is directly related to how the companies have a set of rules on cybersecurity that explicitly sets what the employees are demanded to perform with their devices. The percentage shows that the employees are not well trained with regards to cybersecurity, which can lead to leak of information in unprotected networks. Funnel et al (2020) also demonstrates that 25% of companies are well prepared for home and mobile working, which indicates that there is a lack of cyber security-framed, written rules that employees should follow.

In order to solve this problem, companies have been providing personal devices that have security systems implemented, employees have been utilizing VPN and authorized software to interact with others and increasing awareness of phishing scams, avoiding any suspicious emails received through company accounts. However, some companies do not have well-built security systems, as they have to depend on external services to provide security into their system. They lack the technology to prevent which attacks happen most frequently and respond to them, which can ultimately lead to fatal results, such as data loss and DDoS. This project will provide an efficient way of analyzing threats that the company is facing, which will allow them to see different types of attacks happening.

The project itself will involve a data metrics dashboard, which is composed of different panels that have data representation. There will be multiple dashboards that the user will be able to see, such as Malware, Network traffic, and phishing. These dashboards will allow the user to see

detailed reports related to the threats and analyze the information. If the dashboard shows critical information, then the user can contact the security team directly and handle the problem effectively. This solution tends to provide better performance of the security system, as the customer and the security provider can communicate with each other instead of only the security team dealing with the situation.

1.1 Background

The methodology used in this project mainly involves Grafana, which is an open source visualization and analysis tool to represent time-series database (TSDB) data into graphs and other kinds of visualizations. Other methodology includes Prototype and Wireframe, as they are used to make simple visualizations of how the dashboard will look like. Prototype and wireframe will involve User Experience to understand how the actual users could interact with the provided technology.

1.2 Related work

The most common system that does similar things to the project is Managed Detection Response. Managed Detection and Response (MDR), is a cybersecurity service provided to other companies, which monitors the system in general, detects any intrusion or attack that is happening to the network or the server, and responds to such attacks [2]. The MDR is very similar to the project, whose emphasis lies on threat detection. The service is able to provide customers information that is related to cybersecurity, such as how much network intrusion is occurring, what the average number of network traffic is, and so on. With the database provided, such as AWS

CloudWatch or Graphite, the company can store the data into a database and utilize it to represent what is currently happening in the customer's system.

MDRs are both custom and generically written, as they depend on specific engines to be built on and need conversion of the data by the program. For example, a company would receive huge amounts of data, convert into CSV file using a custom build program, upload the converted data into existing database, and utilize that database to be implemented to open source like Grafana or build a custom website to show the result. This would mostly be a better fit for customers, but MDRs lack in general user experience as it can be difficult for the customer to understand what's really being represented. Without proper user research and designing process, MDR has low potential of being effective as the customer will not be able to comprehend the data being represented to them.

Another similar system includes Managed Security Service Provider (MSSP), which acts similar to MDR [3]. However, MSSP reacts differently as it only monitors network security controls and sends alerts when certain behaviors are observed. Because it is unable to deal with false alarms and the actual threat happening, the IT department of the customer side has to investigate the data and determine that the threat is real, and solve the problem on their side. MSSP is unable to solve the actual problem when the threat occurs because its main purpose is to provide general security service that can prevent the attack, not handling it during the attack. The project itself will provide better performance compared to MSSP, as the dashboard will be able to eliminate any false information and categorize them and the customer can investigate the threat through that dashboard, understanding the problem and directly reaching the security system providers.

2 System Design

High level architecture: The system in theory should have a program that is able to collect, analyze and report on log data. After the data is collected, then they are converted to CSV files, which each threats have special cases assigned to them. Those special cases will define categories that the values are assigned to, which those categories can be used later on to provide detailed information. After the data are transformed, the CSV files then can be uploaded to the database that the project is using. Some of the databases include Graphite or InfluxDB, which allows users to upload the CSV files. The critical step that needs to be taken between the database and the conversion of the data is a virtual cloud that is able to connect both of them. In this case, the AWS S3 bucket acts as a storage where it's able to store the CSV files, and in real time sends the data into the database and updates it. After the database has been set up, Grafana comes into action where it allows developers to choose the database they are using and use the tools to visualize them. The developer can choose which visualizations they want to use, such as graphs, stats, or bar gauge, and represent it visually.

Difficulties: The base of the project requires some cybersecurity data that can be utilized. The original intent was to use public resources which were classified by different threat types (malware, host, fraud, etc). I was able to collect the CSV files accordingly, but was not able to use it due to my lack of skills. Since the data is not real time but a record of a specific time period, it was unnecessary to use AWS S3 bucket as a storage for all the data. Instead, it was much approachable for the CSV files to be directly uploaded to the database.

However, as I was dealing with the CSV files it came to my attention that I wasn't going to be able to upload them to the database. Some CSV files had too many attributes to be initialized, or too small attributes that would not fully serve as suitable data for the project. Furthermore, for each

specific CSV file they had to be transformed into line protocols, which meant that every single file required different scripts to be written. Since it was not possible for me to deal with the complexity, it came to my decision that I would be utilizing a test database that Grafana provides as a default database for the project. It is able to produce sample data such as slow query logs, random walk, predictable pulses, etc. that can be utilized as the source for visualization.

The design process had to be put into, as the dashboard mostly provides information to the users and they should be able to understand what they are seeing. In order to do so I've created wireframes and prototypes of the design for each dashboard, having brief descriptions of what the purposes are for specific panels and what they are representing in general. For example, if the panel was related to a graph that represented logs received by different devices, the user would be able to identify right away with the title and the description of the graph, along with the details involved. I utilized User Experience skills that I've gained from a course taken previously to implement the design.

2.1 Procedure

2.1.1 Design Process The design for the dashboards and panels had to be decided first before the actual implementation. I was able to look at different designs of MSSP dashboards, which most of them have included graphs and other visualizations to summarize the result. Some designs were difficult to analyze as they have only included visualizations, which made it impossible to figure out what they actually represented. Other designs tried to represent the result into a single page, forcing the user to deal with uncategorized results. In order to avoid the problem, it was necessary to apply user experience elements to the designing process. The wireframe was best suited for the design as it held core elements (categorization, threat details) that would be represented by the dashboard.

The wireframe starts with the list of different dashboards categorized by the types of the threats. This way, users can easily access which threat type they wish to analyze. The description for the list is positioned above, while the description for the dashboard itself lies below the list. The description of the dashboard will explain what's happening in the dashboard and provide information related to the specific panels. The images represented as tables and graphs represent the summary of the threat data, which can vary depending on what the threat is. Some panels would include graphs that represent the trend of the threat, while other panels would include log tables that provide detailed information based on time. The following image illustrates the wireframe for the analytics dashboard.

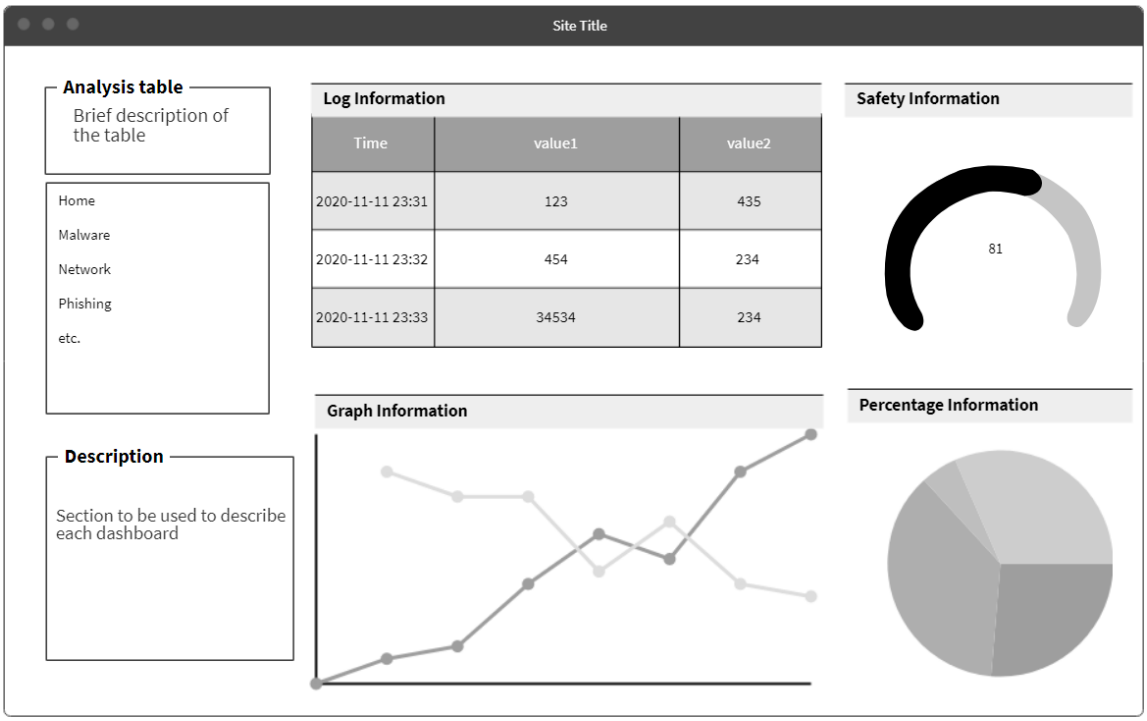

Figure 1: Detailed version of the wireframe that represents sample dashboard

2.1.1 Implementation After the wireframe was initiated, it was ready for Grafana to implement the design. First I created a main dashboard that would represent core information that sub-

dashboards would include. For example, Figure 2 shows the client usage information and log details for Network Traffic dashboard, and graph detailing the received logs for Malware dashboard. From the home dashboard, the user is able to access dashboards with different threat types accordingly and the dashboard list will be available for other dashboards so that the user can be accessible to all dashboards. At the bottom left corner exists the description of the dashboard.

Figure 2: Main dashboard of the system, showing major information related to each dashboard

Figure 3 shows a malware sub-dashboard, which includes information about the malware detection, types of malwares detected, and the frequency of the intrusion. Each panel is able to represent different details related to the threat type, which the developers can edit so that the dashboard contains more relevant information or the dashboard meets the requirement that the client side wishes to see. For example, figure 4 shows a detailed version of the panel where the developer can specifically choose which visualizations that he wishes to use, choose the calculations and variables that will be utilized for the panel, and even apply overrides. Many variations are provided to the developer, allowing him to create different visualizations for

different categories. The panels have titles so that the user sees what those panels correspond to, and are provided with detailed information at the bottom left corner.

Figure 3: Dashboard corresponding to malware

Figure 4: Detailed version of the panel, which the developer can choose which visualization to represent the data

I was able to add several more sub-dashboards that worked similarly to the malware sub-dashboard. Figure 5 represents another dashboard with threat type being Network Traffic, summarizing the trend and the usage information. Adding more sub-dashboard completed the project, successfully developing the security analytics dashboard for the user.

Figure 5: Network Traffic dashboard showing details

2.2 Results

After the build was finished, I wanted to test different users who may use the dashboard. I was able to get my friends and family to try out the dashboard, then asked them how they felt about the project. Most of them were satisfied with the detail that it was providing, as they were able to interpret the data that was represented. They were also positive about the list, as it provided them efficiency to see different results for the threat types. Each individual was able to analyze the panels in each dashboard, and provided feedback that more visualizations could be helpful. They

also mentioned that more elaboration of each panel would be beneficial, as the description was sufficient enough for them to understand.

3 Conclusion

In conclusion, I was able to create a system that summarizes the threat data received to the user, which the user is able to analyze the data without confusion and notify the security team about any suspicious trends. The incorporation of user experience to the design of the project was able to solve the issue related to the understandability and flexibility of the system, providing an efficient way to communicate between the user and system. The dashboards were able to visualize different threat data types, which involved using visualizations such as graphs and tables. The dashboards were included with panels that represented different categories that were related to the threat data types, which could be traffic usage, number of logs, log details, etc. With the finished system it allowed for the users to effectively use the MDR dashboards, interpreting the visualizations and being able to notice any critical phenomenon on site.

3.1 Future Work

For future work, I'd like the dashboard to allow the user to create their own dashboards instead of the developers doing it. Users can ask the developers to add certain panels or dashboards, but if they had the opportunity to create one and have the understandability to handle the elements that belong to the threat types, then the user will build the dashboard more effectively to understand it. Furthermore, there could be more database utilized for the dashboards. Grafana allows you to use multiple databases, so if you want to distinguish the threat data by different databases, you can simply send those data to databases and use a single one for that certain threat type. This will allow more efficiency when building the dashboard.

REFERENCES

[1] Furnell, S., & Shah, J. N. (2020, August 1). Home working and cyber security – an outbreak of unpreparedness?. Computer Fraud & Security, 2020(8), 6 - 12.

[2] What is Managed Detection and Response? Definition, Benefits, How to Choose a Vendor, and More. (2020, September 29). Retrieved November 7, 2020, from https://digitalguardian.com/blog/what-managed-detection-and-response-definition-benefits- how-choose-vendor-and-more

[3] Miller, M. (2020, February 18). What is an MSSP (Managed Security Services Provider)? Retrieved November 7, 2020, from https://www.beyondtrust.com/blog/entry/mssp-managed- security-services-provider