University of Zurich[UZH]

MASTER PROJECT – Communication Systems Group, Prof. Dr. Burkhard Stiller

SHINE: a Collaborative System for Sharing Insights and Information of Economic Impacts of Cyberattacks

Chao Feng, Qiaowen Wang, Xianxiao Xu
Zürich, Switzerland
Student ID: 19-763-796, 19-760-545, 19-763-606

Supervisor: Muriel Franco, Christian Killer
Date of Submission: May 3, 2021

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

ifi

Master Project
Communication Systems Group (CSG)
Department of Informatics (IFI)
University of Zurich
Binzmühlesasse 14, CH-8050 Zürich, Switzerland
URL: http://www.csg.uzh.ch/

Abstract

Cyberattacks continue to post threats to the network infrastructure and the various applications built on it. Numerous organizations and sectors are seriously intimidated by cyberattacks and have suffered severe financial losses. Therefore, the urgency of needing to adopt cybersecurity strategies to reduce potential risks caused by cyberattacks has emerged. In fact, no single organization has the ability to solve the entire problem. Information sharing is a powerful weapon that allows stakeholders to leverage the insights from the whole security community and reinforce their cybersecurity abilities. On the other hand, the economic impacts analysis of cyberattacks could be an efficient way to support an organization's cybersecurity investment decision-making process. Thus, a collaborative platform for the entire cybersecurity community to counter cyberattacks is eagerly required.

This paper proposes the SHINE solution, that is, a collaborative system for information sharing and economic impacts analysis of cyberattacks. In this project, we designed and developed this SHINE platform where interested partners could access a web interface to share their findings, obtain insights on cyberattacks datasets, and analyze the economic impacts of attacks on their business. Network captured data, incident-related data, and economic impacts data could be shared with other users in the SHINE platform. End-users could get economic analysis and incident statistics from their own cyberattack datasets or from the information shared by other users. The SHINE provides hierarchical views so that users are able to get the analysis results of interested data from different depths. The platform was evaluated by three case studies based off two users with distinctive requirements and motivations. The results showed that our platform worked properly for different users with various purposes.

Acknowledgments

We would like to express our sincere gratitude to our main supervisor Mr. Muriel Franco for his sustained support and guidance during this project. Muriel's great knowledge on cyberattacks and security economic analysis has been a great help and inspiration for the project.

Additionally, we want to thank Mr. Christian Killer for his crucial advice that helped us prepare the presentation. Besides, we want to thank Jan von der Assen for helping us build the basic environment of DDoSGrid.

Finally, we want to thank Prof. Dr. Burkhard Stiller for giving us the opportunity to carry out this project at the Communication Systems Group. It is a pleasant experience to work on this project at the CSG.

Contents

Chapter 1

Introduction

Cyberattacks is offensive maneuver targeting at computer information systems, infrastructures, computer networks or personal computer devices [31]. Nowadays, as digital technologies play an increasingly essential role in business and society, the increase of cyberattacks shows an upward trend in different sectors [19]. And these types of attacks could pose threats to confidentiality, integrity, and availability of an organization, and thus, cause huge financial losses [1]. Therefore, an approach that could help the stakeholders to understand the detailed cybersecurity-related information and support the cybersecurity decision making is urgently required. This chapter first gives a short introduction to the status of cyberattacks and then describes the need for having a collaborative approach as a tool to support information sharing of postmortem analyses of cyberattacks among stakeholders.

1.1 Motivation

In the current time, the number of cyberattacks continues to rise in various areas. Among these attacks, money-purpose attacks, such as Distributed Denial-of-Service (DDoS), Ransomware for extortion, and phishing are in the majority. According to a report from McAfee, the estimated global losses from cybercrime escalated to approximately US$1 trillion in 2020, which is nearly double of that in 2018 [35]. Therefore, it is urgent for companies and organizations to adopt cybersecurity strategies to reduce the risks of potential losses caused by cyberattacks.

Since an efficient cybersecurity strategy usually covers several aspects, such as the protection level for the business profile against the most common cyber threats and the employees‘ knowledge of handling and getting rid of cyberattacks, it is crucial for developing strategies through a comprehensive analysis of cyberattacks and their behaviors to reduce the generated impacts as much as possible. Consequently, the importance of having tools to investigate malicious network traffic has been raised. And conducting a postmortem analysis on log files and network trace records after the attack is one of the possible solutions.

However, there is a fact that no single organization has visibility over the entire problem space, which makes the collaboration and information sharing between different users critical. At the same time, collaboration and information sharing are beneficial for both collective resilience and collective actions against cyber threats. Currently, though there are already multiple efficient tools and approaches for cyberattacks investigation, the information sharing on impacts as well as insights of different types of cyberattacks is still challenging [23, 15]. Therefore, possible collaborative solutions supporting stakeholders to share economic impacts or insights of cyberattacks among partners with similar demands and threats are necessary to help with this situation.

1.2 Description of Work

This work illustrates the conception, architecture, design, implementation, and evaluation of a collaborative platform for sharing insights and economic impacts of cyberattacks. The information being processed includes different technical aspects of attacks and economic-related information such as various losses generated uploaded by the stakeholders and economic metrics calculated by the system. The system is intended to input, process, and visualize the relevant information of different cyberattacks concerning the interested business sector, and allow users to share their information regarding a specific cyberattack dataset with other users. The system is designed upon the architecture of DDoSGrid, which is an application for DDoS attacks visualization [10]. The goal of the platform is to help stakeholders, for instance, network operators and decision makers, to access or share insights of cyberattacks data including different analyses and information on cyberattacks datasets and other types of inputs. For example, decision makers of a company might want to evaluate the investment against a specific cyberattack, a researcher might want to compare different cyberattacks in a specific business sector.

Based on the DDoSGrid and other relative researches, several features and economic metrics are applied for the visualization of different dimensions of cyberattacks. New modules and components for generating insights on both economic-level and feature-level are added to the previous DDoSGrid system. Economic-level insights provide information about the economic impacts and cybersecurity measures input by users and feature-level insights provide several selected attack features extracted from the attack information input by users and from the network capture logs (*e.g.* PCAP files). Additionally, as the input data is extended from log files only to other types included, interfaces that allow users to upload and share various sort of data is built as well. Then subsequent to the implementation of new modules and components, the system is integrated with the DDoSGrid system. And finally, the evaluation of the system is run to assess the functionality of the final system.

1.3 Thesis Outline

In the first chapter, an introduction about the motivation and goal of building the system is given. In chapter two, the research on relative literature and project-related work

in several areas, such as information sharing of cybersecurity and economic impacts of cyberattacks, are documented. Chapter three introduces the design of the SHINE platform, including the elicitation of the requirements, (*i.e.* defining the use cases) as well as the design of the architecture, components , and interface mock-ups, and investigates the feasibility and necessity of different modules in the system. In chapter four, the implementation of the system based on the designed prototype, concerning the data model, back-end, front-end, and how the system integrating with existed DDoSGrid is described. And following the chapter five executes an evaluation. And lastly, chapter six is a summary that makes conclusions for the project and suggestions for future work.

Chapter 2

Background and Related Work

2.1 Economic impacts of cyberattacks

Cyberattacks such as malware, web-based attacks, distributed denial-of-service (DDoS) and phishing contribute to huge financial impacts on a business. For these impacts, Anderson et al. categorize them into three components by a framework, which are direct losses, indirect losses, and defense cost [2]. In another framework, Rodrigues et al. describe the different steps and information required when analyzing the economic impacts of threats in a business [38]. According to a report from Accenture, the average annual cost of cybercrime by consequence of the attack for an organization in 2018 rose up to about 13 million USD [7].

To protect oneself from cyberattacks, investment on cybersecurity might be necessary for an enterprise. But not like other investments, the security investments only save costs by minimizing the damage caused by attacks instead of generating monetary returns [16]. Thus, how to measure the economic impacts of these cyberattacks and how to support decisions of investment on cybersecurity seems to be essential questions. Due to that, several models and methodologies have been built to quantified risks or different types of financial impacts based on past attacks and investments.

Gordon and Loeb present an economic model, which is known as the Gordon-Loeb model, that determines the optimal investment amount to protect a given information set based on the vulnerability and potential loss [25], and it is generally used in the field of information security investment analysis [48][32]. Return on Security Investment (ROSI) [41][27] is another benchmarking approach for assessing the investment on information security that have evolved from Return on Investments (ROI) model. Cremonini and Martini extend the ROSI models to Return-on-Attack (ROA) to evaluate the cybersecurity investments from attackers perspective [12]. Similarly, quantitative risk metrics such as Net Present Value (NPV) and Internal Rate of Return (IRR) could also be applied with ROSI to assess the risk of cyberattacks [28]. Furthermore, Bonjanc and Jerman-Blazic evaluate optimal security technology investments and support decision making progress by using a quantitative model to compare quantifications of different security measures [29].

In addition, some multi-criteria decision making approaches are applied in practice due to the purely quantitative methods are not comprehensive enough to estimate every type of losses[1]. Rees et al. describe a decision support system which can calculate the uncertain risk for an organization under cyberattacks by using the approach of Value-at-Risk (VaR) and search for the best combination of countermeasures [37]. Bodin et al. apply the analytic hierarchy process (AHP) to help decision makers to understand and budget the cybersecurity issues [8]. Tallau et al. present a practical application of the Balanced Scorecard method in order to evaluate information security investment decisions for an organization [42].

Relatively, integrated mathematical models like Jerman-Blazic's could measure the security investment more directly, comprehensively, and efficiently, and the model could compare measures intuitively with quantification and thus easily translate into recommendations of security investment [29]. Thus, we decide to design our platform refers to this quantitative model.

2.2 Cybersecurity information sharing

When dealing with cyber threats and incidents, target companies or organisations may possess the resources and knowledge to confront and recover. Nevertheless, lack of communications between targets, the ability of prevention and mitigation can be severely damaged. Considering the knowledge exchange and collaboration among adversaries are a common phenomenon, sharing threat information among targets may be a propitious way to combat this situation as well as for addressing cyberattacks and mitigating their effects [33]. Sharing cyber threat intelligence not only reinforces the defensive capability, but also reduces cybersecurity investment costs [49].

There are several existing consortiums dedicated to solving the problem of cybersecurity information sharing, such as the FS-ISAC, MM-ISAC, T-ISAC, CTA (Cyber Threat Alliance), and Computer Incident Response Center for Civil Society (CiviCERT) [4]. FS-ISAC, MM-ISAC, and T-ISAC are three Information Sharing and Analysis Centers that serve the Financial sector, Metals and Mining sector, and Telecommunication sector respectively. The CTA is an organisation that allows near real-time, high-quality cybersecurity information sharing among their members [13]. The CiviCERT is a network consisting of Computer Emergency Response Teams, Rapid Response teams, and independent Internet Content and Service Providers who help the civil society prevent and address digital security issues [11].

The content of sharing and the expressing method are two non-negligible factors that affect the sharing efficiency. [18] describes Indicators of Compromise as the easiest actionable cyber threat intelligence attributes, which are broadly used in detecting threats. However, cybersecurity information sharing not only involves identifying threats but also involves how to weaken the adversaries and the impact of a cyber incident on an organization.

Bianco et al. [6] organize the threat intelligence into two categories: Technical data and Business data, based on the damage it can cause to a company's reputation by sharing

with the public [39]. The work proposes the Pyramid of Pain [6] to categorise technical data into six levels with respect to the pain one can cause to the adversary by using such information. The category starts from the simplest indicators which can be easily changed (*e.g.,* Hash Values, IP addresses) to tools and Tactics, Techniques and Procedures (TTP) of adversaries.

Furthermore, [34] defines the "actionable information" in IT security as the information that can be used to take actions that mitigate against future threats or help address existing compromises. The taxonomy of information defined in [34] extended the Pyramid of Pain by aggregating six levels into two levels and adding low-level raw data (machine generated, *e.g.,* logs) and strategic reports (highly summarised threat analyses) as two new levels. The two levels in this taxonomy that summarised Pyramid of Pain are potentially considered as actionable information, namely detection indicators, and advisories.

In addition, cyber-threat indicators and defensive measures were the two aspects that were identified based on their destructive power to an organisation's businesses, as the information which can be shared among defenders by Cybersecurity Information Sharing Act (CISA) from 2015 [33]. In [33] seven major types of cybersecurity information identified by [24] are combined with previous two categories. According to [33], cyber-threat indicators include incidents (details of a cyberattack), threats (weaknesses that can be exploited), and vulnerabilities (potential serious issues). Defensive measures include mitigation of cyberattacks, situational awareness (information to handle an incident), security best practices (information on best actions against attacks), and strategic analysis (utilizing information to develop effective defensive measures).

Zheng et al. [49] also distinguish two types of information, specific technical threat indicators which are aligned with technical data in [39], and context threat intelligence which is similar to cyber-threat indicators mentioned before. In [30], the major types of threat information considered are indicators, TTPs, security alerts (advisories, bulletins, and vulnerability notes), threat intelligence reports, and tool configurations (recommendations for setting up and using tools that support the automated collection, exchange, processing, analysis, and use of threat information).

To describe threat details with its context information in a standard manner to facilitate information sharing, Structured Threat Information eXpression (STIX) [5] is developed by MITRE Corporation which serves as an informal ontology that covers a very wide range of security information [34]: from indicators of compromise to contextual information regarding threats, high-level concepts like threat actors and impact information [34, 22]. Within STIX, the stixVocabs is a promising resource to be used to express security information and provide insights into large-scale effects of cyber threats and incidents [22] in an enumeration way.

[24] also discovers two types of policies for exchange, voluntary exchange policy and mandatory disclosure policy. According to [24] the voluntary exchange policy was seen as the richest and most valuable exchange that exists in the cybersecurity ecosystem. But there are still conflicting ideas about which policy is more suitable for information sharing.

2.3 DDoSGrid

Distributed Denial-of-Service (DDoS) attack has been one of the most serious types of cybersecurity threat against the internet operations. Researchers find that network traffic log analysis and visualization tools could be powerful applications to discover patterns and gain insights for cyberattacks. Therefore, Jan and Luc developed a platform, named DDoSGrid [10], to analyze captured DDoS attack logs, extract attack features, and visualize technical metrics.

DDoSGrid [20] was initially developed as a Master's Project of Communication Systems Research Group at the Informatics Department of the University of Zurich. It is an educational and research-oriented prototypical platform for DDoS analysis and visualization. The main objective of this project was to automatically extract relevant metrics from network captured data and visualize these mined features for analysis of DDoS attacks. For example, a researcher could upload a PCAP file that is associated with a DDoS attack, and use the DDoSGrid platform to dig up several levels of network-related metrics, such as the duration of the attack, number of attack packets, and number of source IP addresses. After data mining, visualization tools will be applied to portray different sides of DDoS attacks.

DDoSGrid consists of three components, miner module, web API, and the frontend. Firstly, the miner module is used for technical feature extraction. And a JSON-based web API is used for the integration of these analyzed features with the front-end of the platform, which provides convenient interaction between the platform and end-users.

However, the dimensions of extracted features and the number of visualization techniques are limited in the first version of DDoSGrid. Therefore, Jan has developed the second version of DDoSGrid, named DDoSGrid 2.0 [3], and provides large sets of metrics and visualization tools to get a deeper understanding of DDoS attacks. DDoSGrid 2.0 is a postmortem DDoS attack analysis and visualization platform as well. Two cyberattack analysis applications have been integrated into this new platform, the first component is ddos-clearing-house [14], which is a DDoS data analysis and information sharing platform. The second suite is the DDoSGrid. This new platform provides a mass number of supplemental functions, such as a user authorization section, extra data visualization tools, and PCAP file exportation unit.

Another improvement comes from the Machine Learning (ML) field. The work DDoSGrid-Mining [9], developed within the Communication Systems Group CSG of the University of Zurich UZH, proposed an ML pipeline to build a cyberattack classification model on the top of DDoSGrid plataform. This model uses the features extracted by the DDoSGrid miner module and applies Random Forest and K-Nearest Neighbor Classification Algorithm to label PCAP files with different cyberattack type. This ML model achieves an excellent result for both time-performance and classification metrics, i.e. precision, recall, and F1 scores.

Nonetheless, economic impacts and information sharing are not included in DDoSGrid as well as its improved platforms, and these gaps are what our project wants to fulfill.

Chapter 3

The SHINE Approach

For the sake of eliciting the requirements of stakeholders and designing the specific architectures, the project applies an iterative and explorative process. For the remaining sections in this chapter, the organization is as follows. Firstly, we give an introduction to the process of the SHINE platform design. It includes the requirements in the forms of user stories, as well as the approach of deciding the scope and details of the platform. Secondly, after investigating the literature, extracting requirements, and discussing with the experts on this area, architecture with a comprehensive explanation of the relative components is introduced. This architecture is initiated from the DDoSGrid and is extended with new modules and components for this project. Lastly, the mock-ups of the system with brief descriptions of related functionalities are displayed.

3.1 Overview

As the goal of the platform is to allow users to share insights of cyberattacks such as economic impacts and attack features, it is critical to determine which kind of economic impacts and attack information to analyse in the SHINE system. Therefore, several user stories of various stakeholders are provided in the following sub-sector. Then, following the determination of metrics for analysis and visualization, we specify the scope and tasks of the system.

3.1.1 Scope of system

Embedded in the area of cybersecurity information sharing, the SHINE system is designed to enable different stakeholders with various purposes to gain and share cyberattack information and insights in an intuitive way. Concerning the survey on related literature, requirements, and existed DDoSGrid system, combining with the analysis of datasets made by Jan and Luc in their project [10], our project extracts multiple attack-related information from the network traffic log files (*e.g.* IP watchlist, attack start timestamp and attack end timestamp) and data uploaded by the user, as well as several finance-related

metrics assessing the cybersecurity measures, such as Return on Security Investment and Net Present Value. Therefore, built upon the DDoSGrid system, the SHINE platform not only includes the main functionalities from DDoSGrid, such as datasets uploading, attack metrics extracting, attack data processing, and visualization, but also extends new modules for other types of data uploading, user profile editing, economic metrics calculation, and insights generation. However, as the SHINE system is a prototype system for cybersecurity-related information analysis, it is not fully market-oriented but considering initial cases with education, business, and research purposes. Thus, the SHINE system only provides limited solutions with basic functionality for proof-of-concept purposes. Additional functions, for instance, economic investment recommendation or information sharing forum, are not considered in the system but listed as future work.

3.1.2 Use Cases

In order to show the requirements of different stakeholders intuitively and find essential tasks for the system, user stories for each type of users, including *Researchers*, *IT employees* and *decision-maker of cyber security*, are created. Following Tables 3.2, 3.3, 3.4 are the use case descriptions for each type of user respectively, while Table 3.1 expresses common use cases for multiple stakeholders (*e.g.,* researchers, decision-makers, and cybersecurity experts). Table 3.2 sketches the common use cases of IT employees. For these use cases, personnel that works for companies and might be interested in different actions, such as understand better attacks and their impacts in their organizations, are considered. Table 3.3 delineates the use cases of research institutions, who might pay more attention on the situation of cyberattacks in a specific sector or in the overall environment. Table 3.4 considers the decision-makers of companies who could influence the investment on cybersecurity. These people concern more about the economic impacts of the cyberattacks against the organization and show less interest in the technical aspects of cyberattacks.

In Table 3.1, some common demands across different roles (*e.g.,* researchers, decision-makers, and cyberattack experts), such as uploading files, information sharing in the system, sign-in or sign-up, updating user profiles, checking insights of attacks, and switching between systems are depicted. Requirements are reflected intuitively in these use cases. For example, as the users need to upload and save their datasets in the system, and then decide which of them to be shared and analyzed, several interfaces are necessarily required in these cases, concerning uploading, storing, sharing, and information extracting functionalities. At the same time, some performance and experience improving requirements are also implied. For example, after logging into the system, users would expect the datasets have uploaded by themselves to be loaded automatically in the system, thus enabling users to decide which dataset to have analyzed and not to upload repeatedly. And the system might need to pre-process the uploaded information to ensure the analysis results to be displayed in a reasonable responding time.

In Table 3.2, the demands of IT employees, especially those recruited for cybersecurity, are described. As these people are assumed to concern more about the technical aspects of cyberattacks in their own organization, interfaces for showing the related results are required. Besides, some additional requirements such as a menu for easier searching

and buttons for switching between different levels of insights are requested to enrich the functionality and improve the performance of the system.

ID	Use case description
UC.1.1	As a researcher, IT employee and decision-maker of cyber security, I want to upload a PCAP file or several PCAP files, as well as fill in the information sharing forms for each dataset, so that I could to have all these information saved and processed by the platform.
UC.1.2	As a researcher, IT employee and decision-maker of cyber security, I want to share my attack information upon the platform through an interface in the system, so that other users in the similar business sector or in similar need could access the data and the insights.
UC.1.3	As a researcher, IT employee and decision-maker of cyber security, I want all my uploaded information to be stored and showed in the system whenever I logged in, so that I could then decide to visualize the interested datasets and features.
UC.1.4	As a researcher, IT employee and decision-maker of cyber security, I want to sign up an account and log in the system, so that I could access functionality of the system by authorization.
UC.1.5	As a researcher, IT employee and decision-maker of cyber security, I want to edit and update my user profile, so that the system could display the business profile and basic user information according to my input.
UC.1.6	As a researcher, IT employee and decision-maker of cyber security, I want to find the analysis in the system after uploading datasets, so that I could find interested insights from the results.
UC.1.7	As a researcher, IT employee and decision-maker of cyber security, I want the system to be able to use under different computer systems, so that I could use the it on different devices .

Table 3.1: Common Use Cases of multiple stakeholders

ID	Use case description
UC.2.1	As an IT employee, I want the system to visualize the attack information in user specific view, so that I could have a deep understanding about the attacks targeting at my organization.
UC.2.2	As an IT employee, I want the attack information showing in user specific view could have a menu to select all the features, so that I could investigate interested metrics and find insights from them.
UC.2.3	As an IT employee, I want the system to be able to switch from different level of depth, such as overview, sector view and user view, so that I could investigate the attacks from different angles.

Table 3.2: Use Cases of IT employee

In Table 3.3, use cases of users with research purposes are referred to. In the cases, researchers tend to be interested in finding a different kind of insights (*e.g.* economic- and attacks-related insights) on cyberattacks on overview and sector levels, and thus requiring the system to organize the insights by different sections as well as by different levels.

ID	**Use case description**
UC.3.1	As a researcher, I want to upload the data from various sectors, so that I could compare features and situation across different sectors rather than single sectors.
UC.3.2	As a researcher, I want the system could select the sectors when showing the insights, so that I could find not only the general insights, but also the detailed insights on a specific sector that I am interested in.
UC.3.3	As a researcher, I want the system could organize the insights in different categories such as economic impacts and statistical information, so that I could investigate the cyberattacks from various views.
UC.3.4	As a researcher, I want the system could organize the insights in different level such as general overview, sector level and user-specific level, so that my research could dig into different depth and cover more aspects.

Table 3.3: Use Cases of Researcher

ID	**Use case description**
UC.4.1	As an decision-maker of cyber security, I want the system to visualize the economic impacts in user specific view, so that I could have a deep understanding about the economic impacts and measures of attacks targeting at my organization.
UC.4.2	As an decision-maker of cyber security, I want to upload the cyber security measures taken against different types of cyber incidents, so that the system could evaluate and compare these measures and provide me with some economic insights.
UC.4.3	As an decision-maker of cyber security, I want the insights and results to be intuitively showed, because basically I am not from a technical department and not familiar about cyberattacks.
UC.4.4	As an decision-maker of cyber security, I want the system to shows the economic-related insights by incident category, so that I could compare the measures in each attack type.

Table 3.4: Use Cases of decision-maker of cybersecurity

Finally, in Table 3.4, the main character is decision-maker on the cybersecurity issues of a company. They are considered to pay more attention to the financial impacts of the cyberattacks and show less interest in the technical part. Therefore, requirements on economic impacts information sharing and cyberattack countermeasures editing seem to be necessary. And extra requirements such as intuition and categorization of the outcome are put up to make the system easier to use for stakeholders.

3.2 Architecture for SHINE

The previous section has defined the scope and requirements of our new platform. In this section, we introduce the architecture which we choose to fulfill this mentioned serviceability and functionality. Firstly, we sketch out the general methods which are used to analyze and visualize cyberattacks as well as their economic impacts. Then, we introduce layers, modules of our new platform and give a detailed description for each component. The system design of the SHINE platform is present in the next subsection, where mockups and prototypes be delineated which suits our architecture.

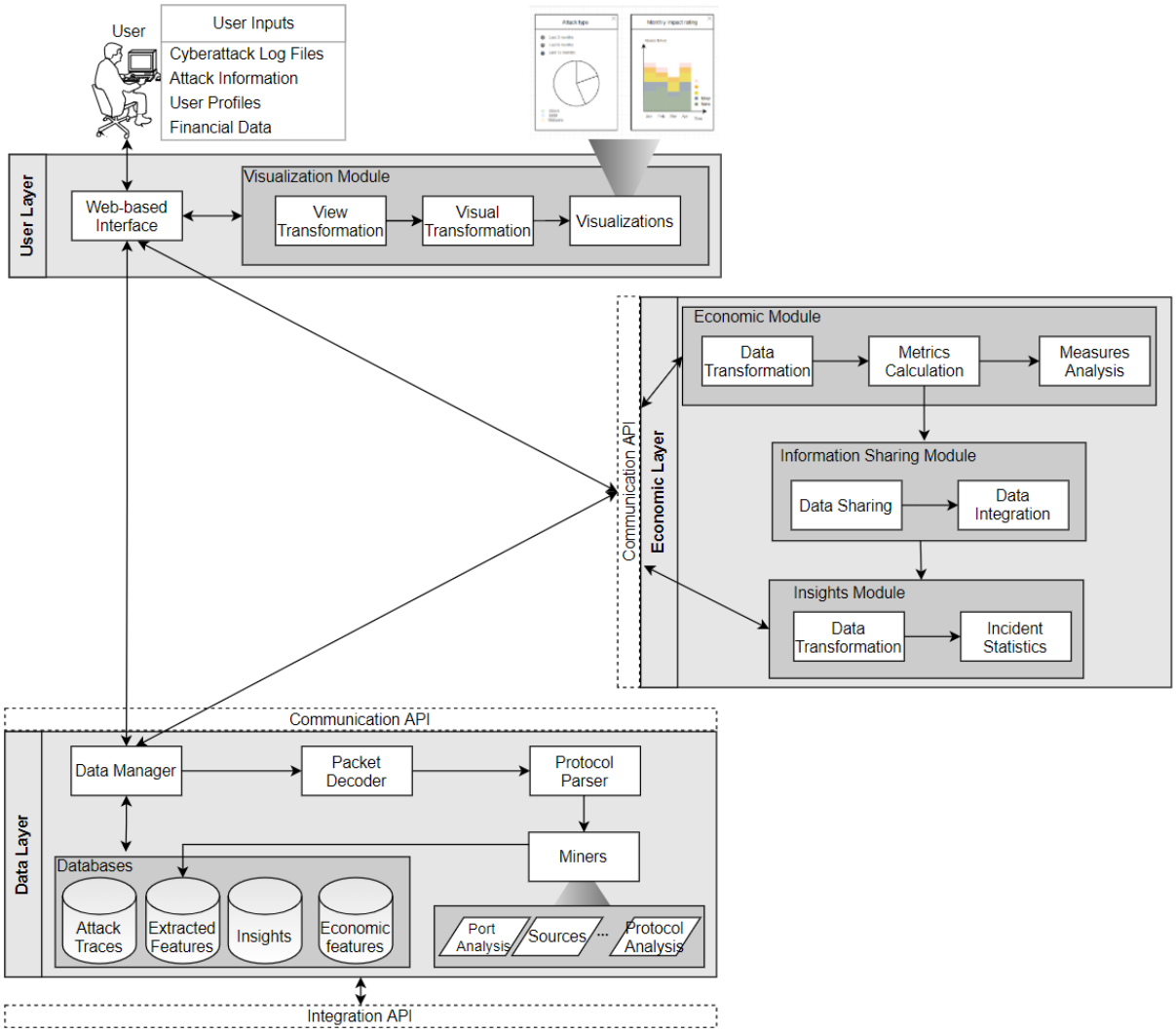

Figure 3.1: Architecture Overview

The SHINE platform can be treated as a new feature which will be integrated into the existing DDoSGrid system, this requires introducing a new layer to the DDoSGrid architecture to provide support for new functionalities. Figure 3.1 visualizes the high-level architecture that we want to build and the interconnection among different modules.

The new layer so-called *Economic Layer* is located between the *User Layer* and *Data*

Layer. Within this layer, several modules that are used to handle different tasks and provide logically separated services, specifically, *Economic Module, Information Sharing Module*, and *Insights Module*. For each module, data flows into different components, transformed, processed, and integrated by those elements. We believe that this layer-module-component architecture could decouple the data from the software function. A new database is used as well to store data related to the *Economic Layer*. The results of this layer will be presented through *Web-based Interface* as DDoSGrid did. The following subsections will outline the purpose of the individual components.

3.2.1 Economic Module

The *Economic Module* is in charge of the finance-related metrics calculation and analysis, including the computation of cyberattack economic impacts, and analysis of cybersecurity measures. As shown in Figure 3.2, there are three components within this economic module, the *Data Transformation* submodule, the *Metrics Calculation* submodule and the *Measures Analysis* submodule. Coming from the *Web-based Interface*, data will be integrated and transformed in the *Data Transformation* submodule, and then it will be fed to *Metrics Calculation* submodule to compute several economic metrics (*e.g,* NPV, ROSI). After that, these calculated results from *Metrics Calculation* will be used to generate analysis through *Measures Analysis* submodule. The final output of this module will be made visible to users via *communication API* and *web-based interface*.

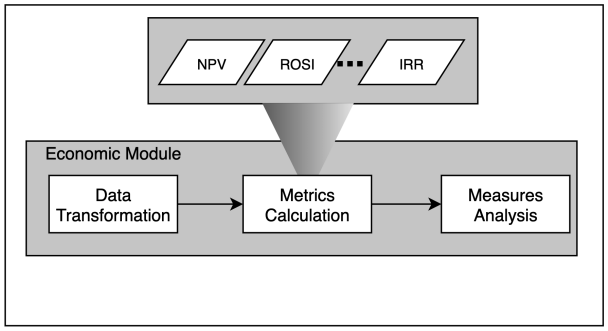

Figure 3.2: Economic Module

The *Data Transformation* submodule is in charge of economic-related data integration, combination, and reformulation. There are three types of economic data, the economic impacts data, the business profile data, and the cybersecurity investment data, and the economic-related data comes from several sources. Economic impacts data comes from every cyberattack dataset and is input by users, which includes all kinds of direct and indirect losses caused by a cyberattack. Business profile data is the financial and accounting metrics of the user, including the annual revenue, cost, and profit information which is feed by users on their personal profile page. The cybersecurity investment data comes from the countermeasure information which is input by the user on their measure page. All of these information and data are collected, combined, and reformed in this submodule and transmitted to the *Metrics Calculation* submodule for economic metrics computation.

As before mentioned, the *Metrices Calculation* submodule is responsible for economic metrics reckoning. Integrated data is used to generate economic figures, like L1, L2, L3, and finally used to calculate the aforementioned Return on Security Investment (ROSI) and Net Present Value (NPV) economic metrics. Eventually, these features will flow into *Measures Analysis* submodule for the inspection of countermeasures of cybersecurity.

Measure Analysis submodule is accountable for detailed examination of the cybersecurity solutions. Ranked by metrics, i.e. ROSI and NPV, these advantages and disadvantages of different measures are visually present in front of decision-makers. This submodule provides a clear sight of security investment and eventually helps support the security-strategy decision making process.

3.2.2 Information Sharing Module

The *Information Sharing Module* serves as the middle-ware for data sharing and data integration between *Economic Module* and *Insights Module* (see Figure 3.3 and Figure 3.1). Data from *Economic Module* will be fed into *Insights Module* after *Data Sharing* and *Data Integration* processes.

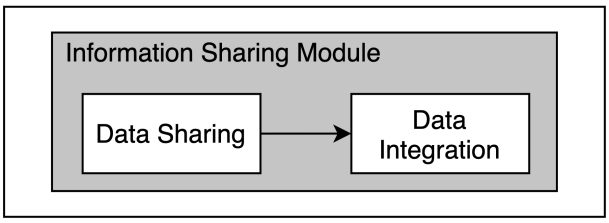

Figure 3.3: Information Sharing Module

For the *Data Sharing* submodule, it controls the sharing of data such as attack information and calculated metrics from the Economic Module. If the user chooses to share his or her data with other users, these attack information and economic metrics will be transformed into *Data Integration* components.

Following that, after the data passing through *Data Integration* components, it is integrated and combined either by sector or user, and then prepared to be transported to the next module. For example, when a user clicks to share the datasets, the *Informaiton Sharing Module* is triggered and data processed by the economic module will arrive at this module, this data is then confirmed to be shared and integrated before it finally proceeds to *Insights Module*.

3.2.3 Insights Module

As stated before, the *Insights Module* obtain input from *Information Sharing Module* as well as the *Web-based Interface* (see Figure 3.4 and Figure 3.1). After the *Data Transformation* process, the input is used to generate statistic information (*e.g,* impact analysis,

asset analysis, adversary analysis) about cyberattacks using *Incident Statistics* submodule. Users can acquire the statistic results through *Web-based Interface*. The results will be stored in a database for further utilization and visualization.

Similar as the submodule *Data Transformation* in the *Economic Module*, this *Data Transformation* is also used for integration, combination and reformulation data which comes from various sources. For example, technical data comes from automated analysis of attack log files, and adversary data comes from users' collection, investigation, and sharing. This module is used to integrate all these data and provide them for the next stage of analysis.

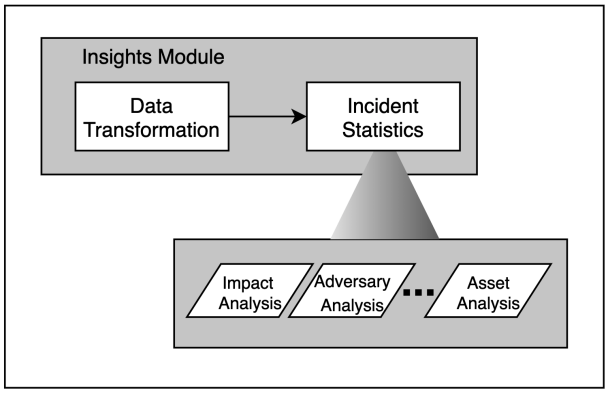

Figure 3.4: Insights Module

The integrated and transformed data passes into the *Incident Analysis* submodule for more specific and in-depth statistics and analysis. This submodule provides multi-dimensional and multi-faceted statistics and analysis, including the analysis of cyberattack impacts, the inspection of attackers' motives and means, as well as the statistic of attacked assets and vulnerable systems, etc. These analyses help users fully grasp their own cybersecurity issues and help users improve their own cyberattack countermeasures. Meanwhile, they can help users get inspired and develop their own network security strategies. Based on these analyses, users can perform targeted vulnerability repairs and security upgrades on their vulnerable applications and systems.

3.3 Mock-ups

In this section, the initial design for SHINE's *Web-based Interface* is presented along with the details of features and possible interactions with the platform.

3.3.1 Sign in, Navigation Bar and Dashboard

When a user enters the application, a home page with a brief introduction of the SHINE platform and the navigation bar which hover on the top of the page will be rendered. *Log-in* and *Sign-up* buttons are provided for a user to redirect to another page to authorize or

register to use the application. Following that, with scrolling down on the page, the user will see paragraphs with more specific instructions about how to use different components and how the platform works to extract and calculate the information that the user needs. The design for *Home Page* will be left out.

The navigation bar (Figure 3.5) extends previous design in DDoSGrid with two additional tabs, *Information Sharing* tab and *User* tab. The *Information Sharing* tab is where the users can obtain economic analysis and incident insights from not only their own data but also data that have been uploaded by other users. The *User* tab is the place for users to alter their personal detail and business profile. The detailed design of the content of these tabs will be explained later in subsection 3.3.3 and subsection 3.3.4.

The Dashboard page is a reuse of the previous design from the DDoSGrid project. It provides users with visualizations of different metrics about the uploaded datasets. After a user uploaded a dataset, the application will process and display the basic information of the dataset. When the user clicks on different icons grouped by various features, a tile will pop up and visualize the selected information.

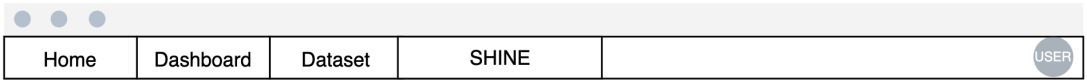

Figure 3.5: Navigation Bar

3.3.2 Information Upload

In addition to the design of the DDoSGrid project, which is uploading, sorting, and comparing different datasets on the platform, SHINE extends *Dataset* with an input form for each uploaded dataset. When a user completes uploading a PCAP file onto the platform by pressing the *File Upload* button, the *Attack Information Form* (see Figure 3.6) will show up by clicking the *SHARE* button to enable the user to fill in the relevant attack information.

As illustrated in Figure 3.6, the *Attack Information Form* includes three sub-groups. The first one is the *Basic Information* subgroup, containing basic information such as sector and organization which the attacked entity belongs to. The second subgroup is *Information Sharing*, including information that could be accessed and shared by an organization, such as impact rating of and impact qualification of the attack. The last part, *Economic Impacts* subgroup comprises parameters on the economic level, such as income loss and corporate income loss due to the attack. After the form is accomplished, the user shall press the *Confirm* button to submit it to the application, otherwise, press the *Cancel* button to cancel the submission.

3.3.3 Information Sharing section

The *SHINE* tab was designed specifically for this extension system. It bears the function of sharing incident statistics, economic impacts, and future suggestions with users. *Overview*,

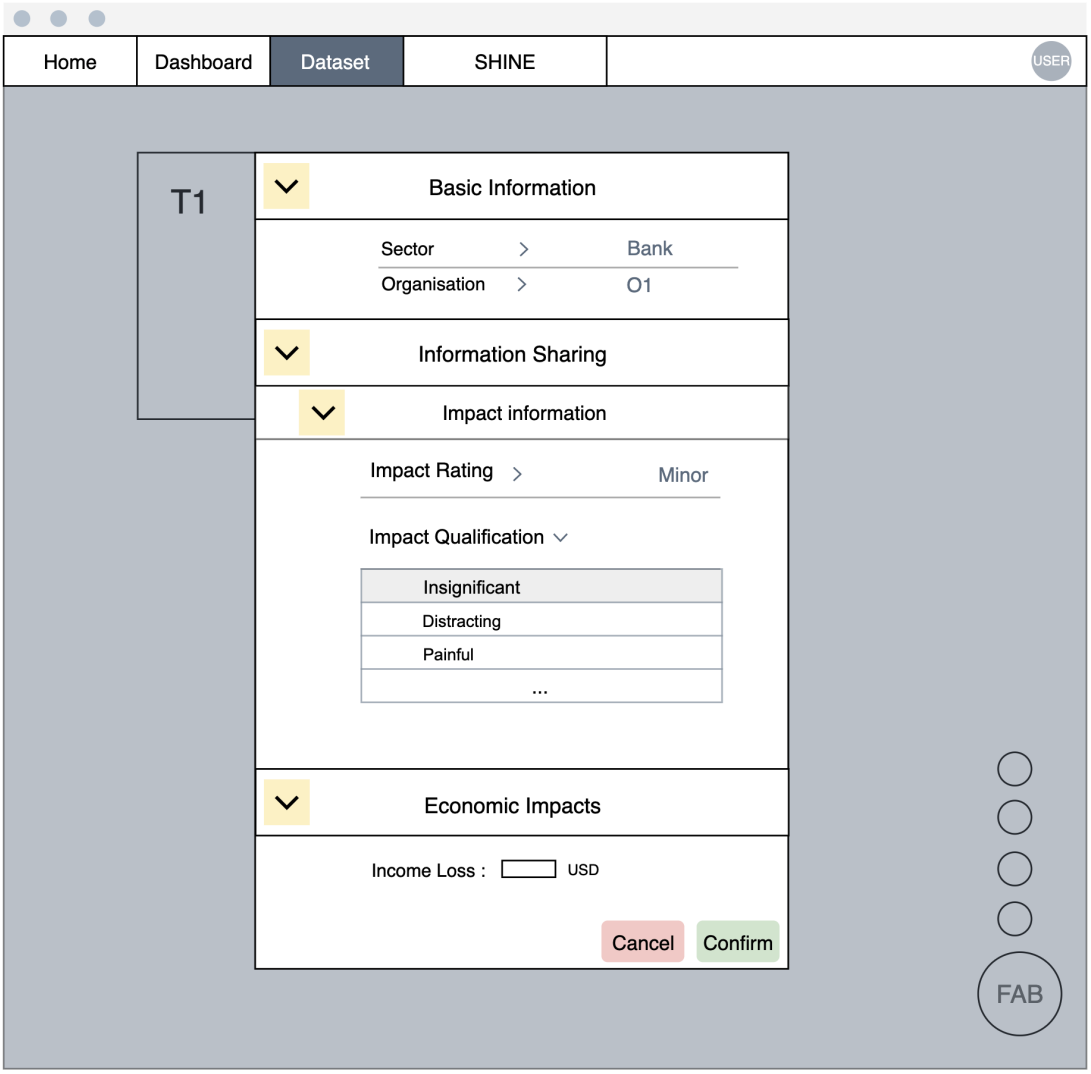

Figure 3.6: Dataset

Sector view and *User view* are three subtabs under the *Information Sharing* tab which target different perspectives of information sharing.

Overview page

To provide users with the most important and the most basic economic impacts and incident statistics, the *Overview* page is used. For example, from the economic view, the *Overview* section provides a bar chart and a pie chart for the financial loss according to time distribution and attack type respectively, and from the incident view, it offers a pie chart and a stacked bar chart to show the distribution of attack type and monthly impact rating respectively. Besides the predefined statistic figures on the overview page, the users are able to navigate to more detailed information either from a sector perspective or from a user perspective through the *more* button. The *more* button will locate users to the specific section in the *Sector view* page and *User view* page.

Figure 3.7: Overview

Information Sharing from Sector View

Detailed incidents analysis and economic analysis from the sector view can be inspected in this section. Users have the full control of which sector they would like to gain knowledge from. As depicted in Figure 3.8, after selecting the Sector view and then *Incidents Analysis* under the *SHINE* tab, below the *Navigation bar*, a *Sector* combo box is given for users to select the interested sector. Two graphs for each of the section are showed on this page by default, and two separate menus for economic impacts and incident statistics are provided as well to ease the information inspection process. After the selection of the element on the menu, the application will process and work out the visualization of the result in a tile. For each tile, there is a specific title on top of it, and the relative graphs appear below it.

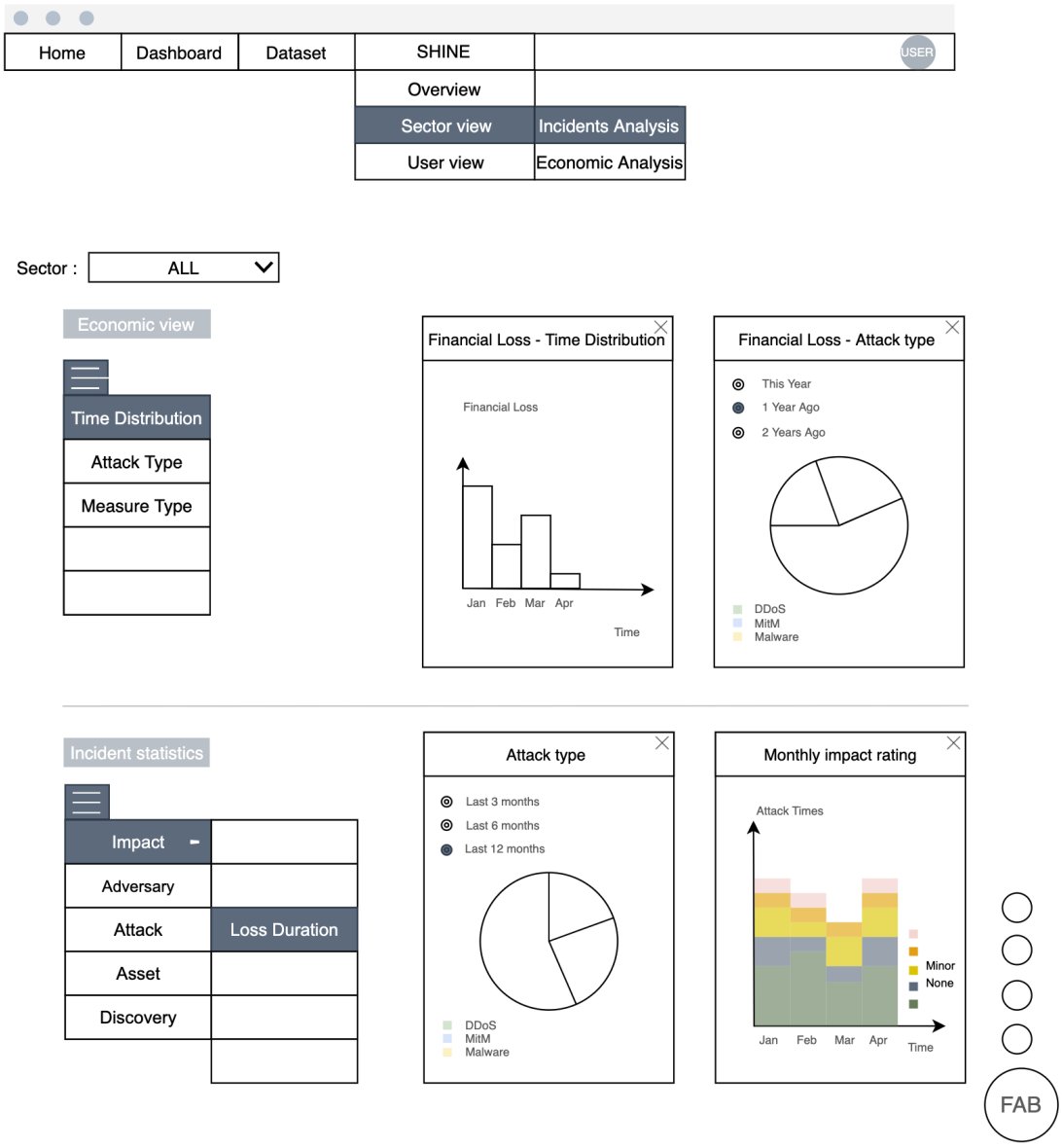

Figure 3.8: Information Sharing from Sector View

Information Insights from User View

Similar to the structure of the *Information Sharing from Sector View*, this section also separates the gained information into economic impacts and incident insights, but both of the results are based on a single user perspective.

After clicking on the *User view* button, the platform will return the organization-specific results calculated from the data extracted from uploaded dataset and forms filled by the user on the *Dataset* page. On the upper side of the page, the *Economic view* part will display different blocks of tables and graphs concerning the measures and relative costs, as well as the comparison of economic metrics such as ROSI and NPV between different measures based on the value of *Attack Type* combo box selected by the user. Following that, the second sub-section *Incident statistic* offers the visualization of different metrics

and features about the attacks against the user's organization. The user selects the interested features on the left menu and the relative results will pop up in a tile on the right. The design of the tiles are similar to the design in the last section as shown in Figure 3.9.

Figure 3.9: Information Sharing from User View

3.3.4 User Profile

The *User* bottom on the *Navigation Bar* will lead users to *User Profile* Page. This is the page where users are allowed to preform modification to certain fields to their basic information (*e.g,* Password, Email), business profile (*e.g,* Revenue, Employee scale) and cyber security measures (*e.g,* Measure type, Initial cost). And the existing users from the DDoSDB system with default user profile data are able to complete and modify their account information in the *User Profile* page as well.

Users could find the complete information by clicking each expands button under different sections and also modify the information in that section. For example, as for *Cybersecurity Measures* section, a user can add new measures by click *Add measures* button. The page then pops out a form including the details of measures for users to fill in (*e.g,* Attack Type, Initial Cost, Annual Upgrade fee). After completing the forms, the form is confirmed and uploaded once the user presses the *Confirm* button, while the process is canceled by pressing the *Cancel* button. And the updated measure could then be viewed in the *Cybersecurity Measures* section.

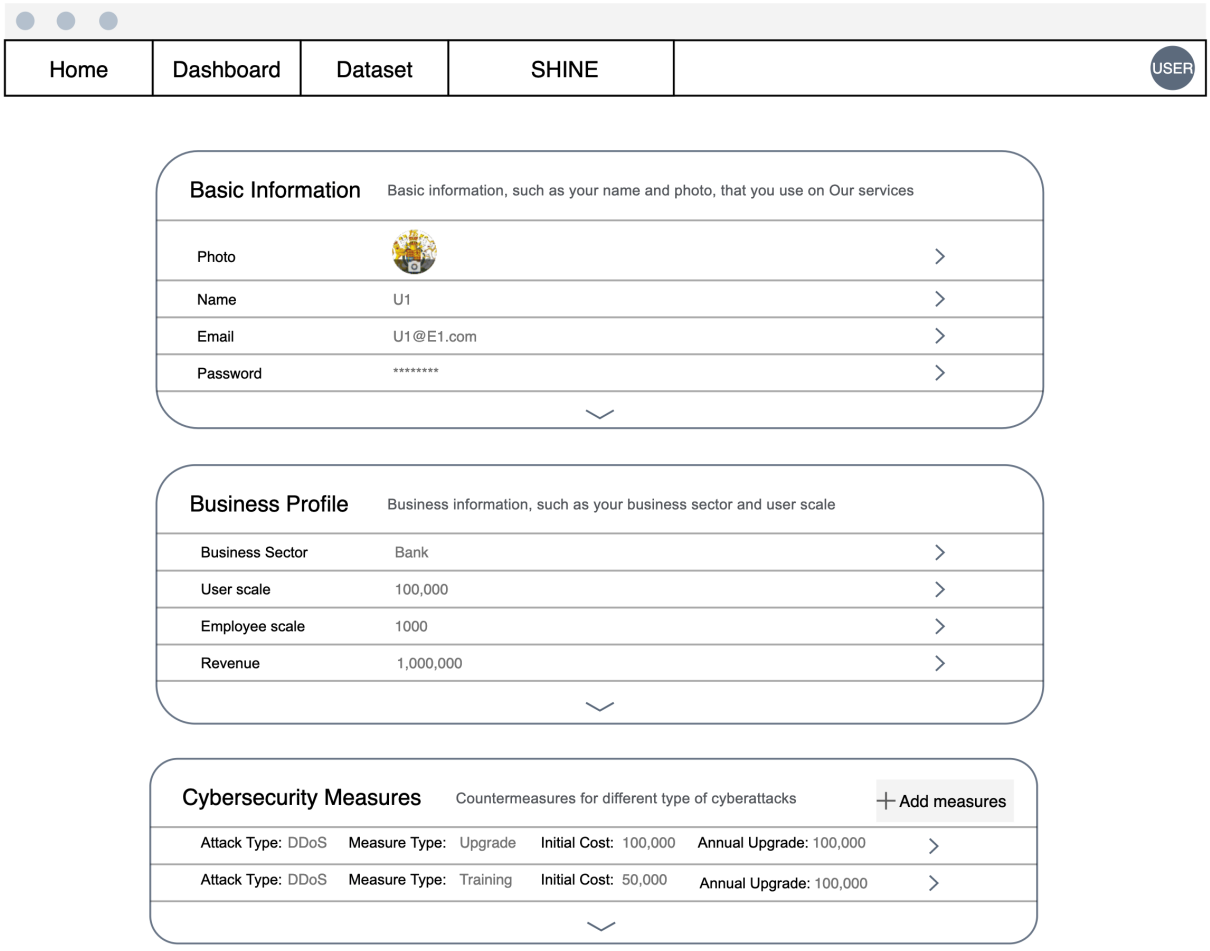

Figure 3.10: User Profile

Chapter 4

Implementation

The previous chapter develops the simplified architecture and system prototype design in accordance with the use cases of different stakeholders. In order to implement the prototype, realization of several aspects such as data models, back-end and front-end of the systems are considered. In this chapter, the report first illustrates how the data organized in the SHINE system, then elucidates the technical details of the back-end as well as the front-end implementation, and finally describes the integration of the SHINE and DDoSGrid systems.

4.1 Data Model

4.1.1 Database design

Based on the SQLite system, a dataset management software, the database of the SHINE system is designed and implemented. To find the necessary tables and fields, after analyzing the requirements of the stakeholders and the information flows of the system in the last chapter, this section outlines the Entity -Relationship diagram (*i.e.* E-R diagram), which represents the relationship of different entity sets and the structure of the database and then gives a short explanation on the diagram. Figure 4.1 shows the E-R model of the SHINE's dataset.

Firstly, for each user of the SHINE system, as they need to sign up and sign into the system, a *UserInfo* entity with *User ID* as the primary key is applied to store the basic user account information as well as the personal business profile information. As the authorization happens in the DDoSDB system, to avoid conflict and to reduce the size of the database, for each user in the SHINE system, the value of *UserID* is exactly identical to the value of the primary key field of the table that stores user-related information in the DDoSDB system. Therefore, the entity *User(DDoSDB)* has a one to one relationship with the *UserInfo*. To store the characteristics of the organization that the user belongs to, an *Organization* entity with a primary key named *OrganizationID* is used and it has been connected to the *User* entity by setting relevant fields in the *UserInfo* as foreign

keys. Similarly, the *Sector* entities store the sector of the business that the user or the organization belongs to, thus has a one-to-many relationship with the records of *Organization* as well as with the records of *UserInfo*.

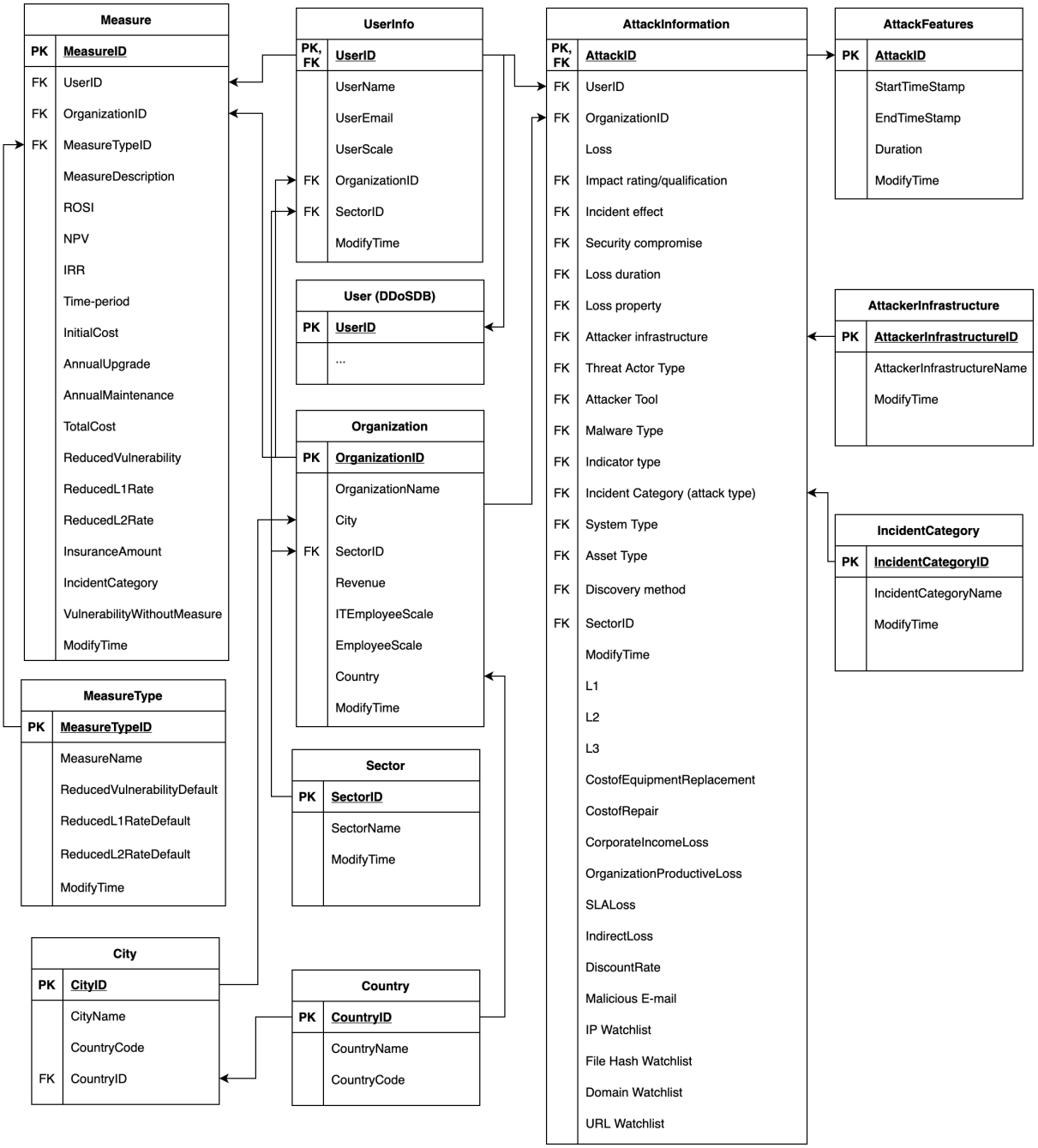

Figure 4.1: E-R diagram of the SHINE system

As for the information related to attacks, for each cyberattack, there is a unique *AttackID* to distinguish it, and the shared attack information uploaded by users (*e.g.* Impact rating, Incident effect and etc.) as well as the financial losses information (*e.g.* Cost of Equipment Replacement, Cost of Repair, Corporation Income Loss and etc.) will be stored in the *AttackInformation* table. By using the *UserID* and *OrganizationID*

fields, this entity has a many to one link with the *Organization* and the *UserInfo* entity respectively. On the other hand, the attack features extracted from network traffic logs uploaded by users, such as Duration of attacks, Start time stamp, and End time stamp of the attack, are characteristics of *AttackFeatures* entity, which has a primary key field *AttackID* and serves as foreign key and primary key of *AttackInformation* entity to link these two tables together.

For the economic impacts, as the system evaluates the cyber security countermeasures by different economic metrics including ROSI and NPV, these metrics and related information (*e.g.* Costs of measures, Attack Type, etc.) are stored in the *Measure* table, which has many to one connection with *UserInfo* table. Additionally, linked through *MeasureTypeID* with *Measure* table, a *MeasureType* table with pre-defined high-level measures data of cyber security, such as upgrade and training employees, is provided for a user to select.

In accordance with the Pyramid of Pain [6], Actionable Information [34] and STIX vocabulary [40], several qualitative parameters are included in the *AttackInformation* table. Among them, for those data fields in enumerate format, such as *Incident Category* and *Attack Infrastructure*, tables with pre-defined data are provided to link with them. For example, the *IncidentCategory* table consists of categories of attacks, such as DDoS, Unauthorized access and Malicious code. The *Incident Category* field in *AttackInformation* table is related with the *IncidentCategoryID* field in *IncidentCategory* table, where user could then request the name of the incident category. For sake of the clearness, only two of the enumerate tables are exhibited on the Figure 4.1, which are the *IncidentCategory* and *AttackInfrastructure*, while other similar tables are omitted.

4.2 Back-end

In the previous section, we have discussed the data we are going to use in the back-end logic. Due to the need of integration with the DDoSGrid platform, we decided to use a decoupled architecture. The development of the front-end keeps the decision made by the DDoSGrid platform and continue using Vue.js. For the development of the back-end, unlike DDoSGrid, we decide to use Python as the coding language and *Django* as the back-end framework because of its powerful built-in functionalities. This section focuses on the concrete implementation of the back-end.

4.2.1 Economic Metrics Calculation

As the economic-related data (*i.e.* economic impacts information and countermeasure information) is fed into the system and stored in the database, the economic metrics are calculated by the back-end system and the calculated results are then gathered in the relative data fields.

Firstly, when the details of the measure is collected into the database, the intermediate variable total cost $cost_{total}$ is settled by adding up the initial cost $cost_{initial}$, the

cost of annual upgrade $cost_{annual\_upgrade}$ and cost of annual maintenance of a measure $cost_{annual\_maintenance}$.

$$cost_{total} = cost_{initial} + cost_{annual\_upgrade} + cost_{annual\_maintenance} \tag{4.1}$$

Analogously, when the information about losses confirmed to be uploaded, the intermediate variables, $l1$ (*i.e.* the losses provoked during the repairing time), $l2$ (*i.e.* the losses generated during the detecting time) , and $l3$ (*i.e.* the fixed losses) are immediately calculated in accordance with the equations as follows, which is $cost_{total}$ adding up several items.

$$l1 = cost_{repair} + loss_{corporate\_income} + loss_{organization\_productive} \tag{4.2}$$

$$l2 = loss_{corporate\_income} + loss_{organization\_productive} \tag{4.3}$$

$$l3 = cost_{equipment\_replacement} + loss_{sla} + loss_{indirect} \tag{4.4}$$

After all the intermediate variables have been figured out and stored, the risks of being attacked by a specific category of the incident will be then worked out as follows.

Assume there are n attacks that belong to the kind of attack that we are interested in. Therefore, the $R0$ represents the overall losses caused by this particular type of attack (see Equation 4.5). As the vulnerability without measure $vulnerability_{without\_measure}$, the reduced rate of vulnerability $reduced\_rate_{vulnerability}$, reduced $l1$ rate $reduced\_rate_{l1}$, reduced $l2$ rate $reduced\_rate_{l2}$ are parameters included in the countermeasure details, which have been uploaded, then the risk after applying a measure (RC) is produced by the Equation 4.6.

$$R0 = \sum_{i=1}^{n}(l1 + l2 + l3) \tag{4.5}$$

$$\begin{aligned} RC = {} & vulnerability_{without\_measure} * reduced\_rate_{vulnerability} * \\ & (l1 * reduced\_rate_{l1} + l2 * reduced\_rate_{l2} + l3) \end{aligned} \tag{4.6}$$

For the next, as the calculation for both of the risks RO and RC have been given by Equation (4.5) and Equation (4.6), the economic metrics, $ROSI$ and NPV, are able to be calculated by the module. The $ROSI$ is the metric that evaluates the return of a security measure, thus it is represented by the ratio of the benefits (*i.e.* the reduced risks minus the total cost) and the total cost (see Equation 4.7).

$$ROSI = (R0 - RC - cost_{total})/cost_{total} \tag{4.7}$$

The calculation process of NPV is a bit more complicated. Since it represents the present values of future cash flows, it uses a loop to calculate the annual number year by year. Thus, to calculated it, we need parameters including how long will the measure lasts in years, which is countermeasure persistent time period T, and the yearly discount rate r. And the final NPV is the sum of the values switched to the present values in each year (see Equation 4.8).

$$NPV = -cost_{initial} + \sum_{t=1}^{T}[(R0 - RC - cost_{annual\_upgrade} - cost_{annual\_maintenance})/(1+r)^t] \tag{4.8}$$

4.2.2 Data Management

The data being transmitted within the SHINE platform is composed of two parts, specifically, the economic part which are predominantly numbers, and the information sharing part which for the most part are single and multiple selections made by users supported by predefined enumeration options on account of the involvement of the STIX vocabulary.

How to manage these options provided to users was a crucial decision to make for the system with respect to the convenience of use and flexibility of management. One option may be to have the administrator directly access the raw data in the database to perform management, but this operation is rather reckless and may cause severe problems to the system. In Order to increase the security of the system, an additional system is required for the administrator to manipulate the data without direct contact with the database.

To achieve the previously mentioned purpose, we could either build an admin system from the scratch, or we could choose to use the *Django REST framework* automatic admin interface. The decision was made after we conduct the comparison between the two approaches. The administrator's responsibility in the SHINE system is limited to addition, deletion, and modification of some certain data, which means we need a simple administration system that supports mentioned functions. This made the Django admin interface exactly what we need, it is able to read metadata from models created based on the data models described in section 4.1 to provide a quick, model-centric, user-friendly interface to manage the content in the database [43] without writing any explicit code for the addition, deletion, and modification processes.

Data Models and Applications

Since we built our database prior to the entire system, we need to map each table in the database to the data model, the *Django REST framework* offered us an efficient command (*i.e. python manage.py inspectdb > models.py*)to auto-generate the models from the existing database to a specified file.

Despite this amazing built-in function, auto-generated models are not entirely trustable, slight changes may be needed. Particularly, the many-to-many relation for multiple-choice fields, which were stored in a separate table with the help of the foreign key fields to establish the relationships, requires to define a *ManyToMangField* in the data models to indicate there is a many-to-many relation between two tables. This kind of field serves the purpose of designating the types and places of the related data fields instead of to actually store the record of the relations, and it is managed by the back-end rather than the database.

After the generation, we created applications that can divide the system into different components to distinguish the services supported by each application. The generated models then are placed into corresponding applications inside the *models.py* files. All applications were then added to the *INSTALLED_APPS* in the *setting.py* as required by the *Django* framework.

Data Management in Admin Site

In the SHINE system, we have 17 models that need to be managed by the admin system, 13 of which are information sharing related models, and the other 4 are user account related models (for detailed information see Table 4.1). Since we use the DDoSGrid system as a base of the SHINE system, we inherited the login function as well. The ordinary user's identity verification and authentication are provided by the DDoSBD system, and the SHINE system stores other information about the user that can be used for information sharing. We used the *UserID* from the DDoSDB system as the primary key in the SHINE system to store the user profiles and to match the records to the user records in the DDoSDB. The administrator's registration and authentication are powered by the *Django admin system* and the relevant information is kept and managed in the SHINE system as the default settings did.

With the aim of utilizing the *Django admin interface*, we registered models which need to be managed by the admin system within the *admin.py* file inside each application, and we created classes within *admin.py* that inherited the *admin.ModelAdmin* class to control the display of the admin site and registered these classes in the same way as we did for the data models. The layout of the pages and some functions within the pages can be managed by assigning values to the inherited variables (*e.g.,* search_fields, fields, list_display, list_filter), we utilized these variables to organize our admin site based on the data models, the relationships among data models and our designs.

It is worth noting that some foreign key data fields in a model are interdependent, for example, the city and the country in the organization model. For the reason that extending the *Django admin interface* is unlike writing an admin system by ourselves, we decided to replace the normal *models.ForeignKey* field with the *ChainedForeignKey* field which belongs to a third package *django-smart-selects* to achieve chained selection in the admin system. One of the advantages of the *ChainedForeignKey* is that it requires minimal code to take effect. We also defined methods to validate the legality of all data passed to the *models.DateField* when it's necessary, and assigned the methods to the parameter *validators* inside the data field.

Application	**Model**	**Admin system management support**
register_application	Application	Yes
attack_features	Attackfeatures	No
attack_information	Attackinformation	No
city_country	City	No
	Country	No
information_sharing	Assettype	Yes
	Attackerinfrastructure	Yes
	Attackertool	Yes
	Impactrating	Yes
	Incidentcategory	Yes
	Incidenteffect	Yes
	Lossproperty	Yes
	Malwaretype	Yes
	Securitycompromise	Yes
	Systemtype	Yes
	Threatactortype	Yes
	Discoverymethod	Yes
measure_type	Measuretype	Yes
measure	Measure	No
organization	Organization	Yes
sector	Sector	Yes
user_info	UserInfo	Yes

Table 4.1: Admin system management support status

4.2.3 Communication between Front-end and Back-end

To support the front-end back-end communication and to keep in line with the decision made for the admin site, we chose the RESTful API as the style to access and manipulate data in the database, and we used the *Django REST framework* to build the APIs for the SHINE platform. The *Django REST framework* can be used for rapidly building RESTful APIs based on *Django* data models [36] and it is able to reduce the amount of code needed to create RESTful APIs.

Serialization and Serializers

Another advantage of the *Django REST framework* is its serialization. To store the data passed from the front-end and to push data to the front-end with *Django REST framework* need the help of the serializer classes. We used two approaches to create the serializer classes.

The first kind of serializer classes inherited the *serializers.Serializer* from the *rest_framework* package and provided a way of serializing and deserializing the instances into representa-

tions such as JSON [26]. Within the serializer classes, the fields that need to be serialized and deserialized require to be explicit defined. The inherited methods *create()* and *update()* defined how fully fledged instances are created or modified when calling *serializer.save()* [26]. Overwrite of the *create()* and *update()* method are recommended for any modifications to the default create and update process. We defined the serializers in this way whenever the queryset that needs to be serialized does not have any predefined data model matching with it.

The second style is to inherit the *serializers.ModelSerializer*. The inheritance of *serializers.ModelSerializer* is a shortcut to create serializer classes, and it made the code more concise [26], but it came with a defect that a predefined data model is a requisite. Consequently, we defined serializer classes in this manner whenever the correlate data model exists. We added extra data fields in the *ModelSerializer* with the explicit specification given our requirements. And due to the nature that default *ModelSerializer* uses primary keys for relationships, in order to acquire a nested representations of the *PrimaryKeyRelatedField*, we assigned values to *depth* to control the depth of relationships that should be traversed before reverting to a flat representation [26], for example, the *Sector* information in the *Application* data model is a *ForeignKey* field, it was connected to the *PrimaryKey* field in the *Sector* data model, if we would like to obtain other information about a sector in addition to the value of the primary key when we query for a *Application* record, we can set *depth* to 1 to meet our demand.

The data storage procedure extracts the data from the sent request then deserialize the *QueryDict* type of data into *OrderedDict* and store the data into the database as a fully populated object instance. The data query procedure involving transformation from *Querysets* to *OrderedDict* using corresponding serializer class. The form of responses forward to the front-end can either be the OrderedDict rendered specifically as JSON using *JSONResponse* or using the *Response* object introduced by *Django REST framework* to take raw *OrderedDict* type of data to negotiate, determine and transform the content to the correct content type. We decided to take advantage of the feature offered by the framework and hand over the responsibility of choosing the data types of the responses to the framework to avoid any issues raised during the process of determining the type of data to be returned.

API view provider

The *Django REST framework* provided two wrappers to write API views, The *@api_view* decorator for working with function-based views and the *APIView* class for working with class-based views [26]. Out of these two wrappers, we chose to use the second one considering the class-based views allow for the reuse of code and the coverage for the common use patterns of querying data is promising.

There are several different view classes available to be inherited, the basic *APIView*, the *GenericAPIView*, five *Mixins*, and nine *concrete generic view* classes which are the most frequently used view classes unless one needs heavily customized behavior [43]. The largest proportion of the business logic of the SHINE platform is to query a collection of instances from the database or create a record in the database determined that the most common

inherited *views* in the SHINE back-end are the *ListAPIView* and the *CreateAPIView* from the *concrete generic views*, and the base class *APIView* which is suitable for creating custom-made query statements, storage process, and URLs.

The *ListAPIView* extended the *GenericAPIView* and the *ListModelMixin*, it works with the data models and the serializers corresponding to these data models to query for a collection of model instances. At the same time, it can achieve the filtering function with fields *filter_backends* and *filter_fields* being specified. Mostly, when the results we require are raw instances of a model without any aggregation operations, as we did for querying the options to multiple selection fields, we would use this class to achieve a rapid implementation. We set *DjangoFilterBackend* as our default filter back-end in the *settings.py* in order to obtain the functionality of filtering results based on given fields of the matching data model. And the *filter_fields* attribute was set to the fields needed to filter against to perform equality-based filtering [26].

Analogously, the *CreateAPIView* extended the *GenericAPIView* and the *CreateModelMixin* which means to utilize this class, data models and their matching serializers are necessary. The *.create()* method provided by *CreateModelMixin* implements creating and saving a new model instance [26] based on the data came with the *request*. If the data being passed to the back-end does not need any adaptation to the serializers, we would directly call *.create()* function in the *post* method to inject the data to the database. If the data requires additional operations before writing in the database, we would overwrite the *post* method following the demands and then call *serializer.is_valid()* function and *serializer.save()* function to trigger the database writing process.

In the SHINE system, the *APIView* were primarily used to handle the bespoke query statements. One of the advantages of the *APIView* is that it allows for maximum customization. And to generate data for the *Incident Statistics* part of the SHINE, tailor query statements to perform aggregation, filter, group-by, and order-by is a must made this freedom of coding vital. The serializers were defined and kept in line with the query results, every field that appeared in the results has a paired field in the serializer classes. The detail on writing query statements will be elucidated in the following sub-section.

Data query

There are two methods to query data for the front-end, one option is to inherit classes provided by the *Django REST framework* and use the default methods defined in the inherited classes like above-mentioned *CreateAPIView* and *ListAPIView* which are able to reduce the number of lines of code, the other option is to rewrite the query statements and overwrite the inherited functions to return customized results as we intended to accomplish within *APIView*. The fact that the data requested by the front-end for the display purpose is relatively complex determined that we need to use both these methods.

The first method is rather intuitive and easy to use, there is no particular need for writing any query statements, the father classes have already implemented the basic query statements. With the setting of the variables, the creation of a record and query for a list of records with a basic equality-based filter can be achieved effortlessly.

To make the data for the front-end ready to use and reduce the data process load of the front-end, complex query statements at the back-end are necessary. With the help of *Django's* support of Object Relational Mapping (ORM), by utilizing the database-abstraction API given by *Django*, we can convert the raw SQL statement to query data using python. The frequently used methods in SHINE for looking up objects are *all()*, *filter()* and *get()*. To return new *QuerySets* with modified(aggregated) results, the most used method that functioning with aggregation function is *annotate()*.

URL registration

In order to provide service for the front-end, the communication URLs are mandatory. Within each application, we created a URLConf and named the file *urls.py* to store all the URLs to be registered. Since all the views that provide service are class-based views instead of single function views, the *as_view()* is compulsory to convert a class-based view to a callable view that could take a request and return a response. The strength of class-based views reveals when we have multiple methods defined within one view class, the *as_view()* function can automatically match map the *http_method* to the correct handler function. This property simplified the URL registration process by assigning one URL to one class-based view to manage all methods inside that view.

For writing the URLs, two functions with nearly identical functionalities can be used inside the *urlpatterns* list, namely, *path()* and *re_path*. We used the *re_path* for it's more powerful with allowing write URLs with regular expression. The URLs written in the form of regular expression were started with a *r* and the variables inside URLs were followed by the regulations to be obeyed. For the filters operating with *filter_backends* and *filter_fields*, it is unnecessary to include the filter fields inside the URLs as variables. The filter fields are optional variables when calling the URLs, they can either be added at the end of the URLs to take effect or be left out, for example, if the *filter_fields* has been set to the field *'is_valid'*, when accessing the URL, one can attach *?is_valid=TRUE* to the end of the URL to filter out the instances with *'is_valid'* field equals to *True* or omit it to skip the filtering process.

After assigning URL to each view class in the *urls.py* file within each corresponding *Application*, the *include()* function from *django.urls.include* were called inside the *urlpatterns* list in the root *urls.py* file to point the root URLConf at each separate URLConf. With the URLs and views being set up, after starting the *Django* service, the view classes are available to be accessed by concatenate the server IP address, port number, root URLConf and application URLConfs.

In this section, we documented the back-end system of the SHINE platform, including the economic metrics calculation logic, the data management system, and the communication between front-end and back-end. Developers and administrators can use this document as a baseline to maintain and improve the back-end system. At the same time, if maintainers want to add new economic metrics calculations, they can also refer to this content. Next, we will focus on the design, development, and implementation of the front-end applications.

4.3 Front-end Development

The previous section has discussed the back-end applications of our SHINE platform, and in this section, we want to present the components of our front-end applications. In the first part, we firstly show the reason of the chosen of front-end framework. Following that, we demonstrate the organization of our hierarchical components. Additionally, we describe how these components are built.

4.3.1 Base Web Framework

The foregoing discussion has pointed out that our SHINE system is built based on the DDoSGrid platform, so that we decide to use the same front-end framework as it. As the starting point of our front-end development, DDoSGrid uses Vue.js [47] for their front-end development. Since that, the best practice for this integration is to use the Vue.js as our front-end development as well.

Vue.js is a powerful and lightweight front-end JavaScript framework for user web-interfaces building and single-page application creation. The most attractive is that it allows the developer to make a dynamic loading component instead of reloading the full page. Besides, the router mechanism supports in-need URL-component matching and loading.

4.3.2 Libraries Choosing

To provide a consistent and unified design for our platform, reusable components and libraries for the front-end elements organization and visualization are required.

Components

As one of the most important design styles, the material design provides a user-friendly and visually unified interface design language. So that, we choose to use a material design styled components library as our front-end components library. Besides, since we used the Vue.js as our framework, a Vue.js supported library is required. We decide to use Vue Material [44] as the components library, not only can this library provide an easy-to-use layout and components, but also prevent us from costing unneeded time on CSS and HTML design and configuration.

Visualization

The goal of our visualization library choosing is to find a simple, and data-option- separated Vue.js chart library. This guides us to the Vue-Chart.js [45]. It is a simple but powerful visualization library, provides a great number of chart types, such as Pie chart, Bar chart, Line chart, Doughnut chart and Radar chart. They could efficiently present

different forms of data. Besides, options, like the colors and labels, are independent of the data, and each diagram could be easily personalized and customized.

Select Field

The information sharing and countermeasure adding applications heavily relay on the selection of input data, which means we need a user-friendly and powerful select field. We need a select component that could handle different demands, such as single select, multiple select, grouping, option filtering, and new item creating. According to our needs, the select element of Element UI [17] is chosen to be our select component. Beyond the basic functions of single, multiple select, Element UI could also create and select new items which are not included in select options.

Sidebar

To make it easier for the user to find specific results, we provide a sidebar with various categories of features, so that the user could have quicker access to a particular result he interested in. This sidebar module must be compatible with our previous components library while providing clear and convenient functions. Vue-sidebar-menu [46] is such an easy-to-use and practical vue-router compatible sidebar menu. As shown in Figure 4.2, this sidebar provides a simple hierarchical menu, and users can quickly find the result page they want according to the hierarchical items.

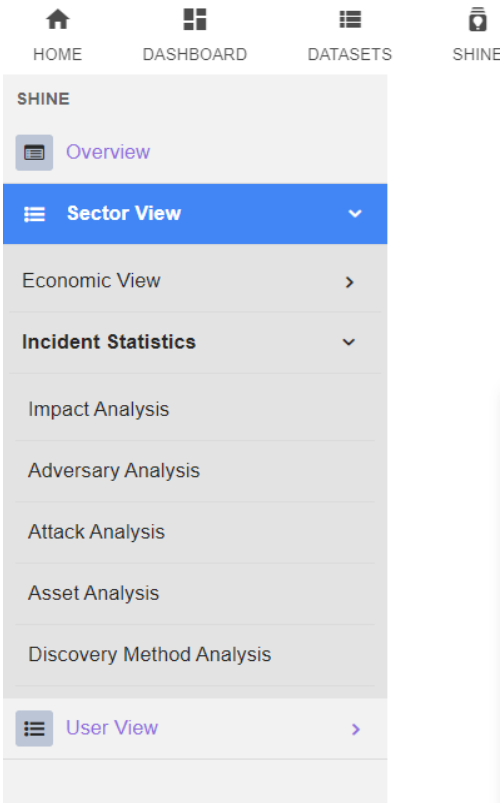

Figure 4.2: Sidebar menu under the SHINE tab

4.3.3 Datasets Page

The *Datasets* page is inherited from DDoSGrid, which is used to upload web load files, list all uploaded and analyzed data sets on the platform. To adapt this *Datasets* page into our SHINE platform, we have added some new features to this page.

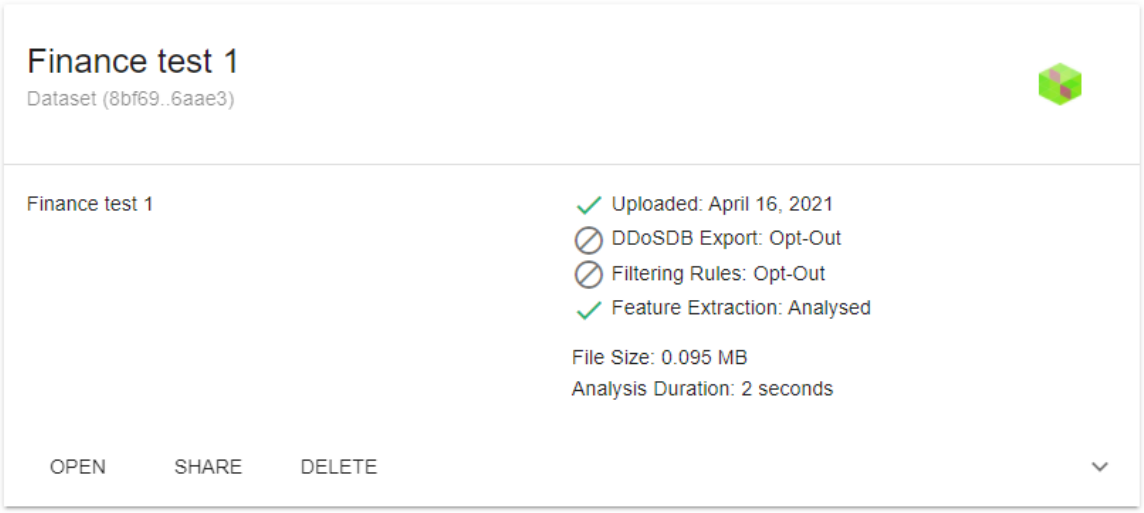

Figure 4.3: Front-end view of Dataset card

After uploaded into the system, a log file is listed in the *Datasets* page. As we can see from Figure 4.3, user could click the *Open* button to visualize this dataset on the *Dashboard* page or click the *Share* button to share this log file with other users.

When clicking the *Share* button, a *Share more information* dialog appears, and the user could fulfill these information to share with others.

Shared item name	**Item description**	**Input type**
Business Sector	The sector that the organization belongs to.	Single select
Organization	The name of the attacked organization.	Single select

Table 4.2: Shared items in basic information

There are three types of information that user could share with other users, including the basic information (see Table 4.2), the economic impacts data (see Table 4.3), and the incident information (see Table 4.4).

The basic information includes the sector and organization information that this log file belongs to. The economic impacts data includes all aspects of cost data caused by this attack. And the incident information included technical, impact, adversary data for this attack. After clicking the *Save* button, this filled information is stored in the database and shared with other users.

Shared item name	**Item description**	**Input type**
Cost of Equipment Replacement	The price of new equipment when the old equipment is failed after an attack.	Text input
Cost of Repair	The price of repair works of employees or external contractors, to eliminate the consequences of the security incident and restore system or service in normal operation.	Text input
Corporate Income Loss	The loss suffered on the revenue side due to system or service failure as a result of the incident.	Text input
Organization Productive Loss	Reduced business productivity due to system or service failure.	Text input
SLA Loss	Loss due to non-compliance with statutory provisions or contractual obligations.	Text input
Indirect Loss	Potentially long-term consequences represent damage to the reputation of the organization, the interruption of business processes, loss of intellectual property, and damage to customer confidence.	Text input

Table 4.3: Shared items in economic impacts

Shared item name	**Item description**	**Input type**
Impact rating	Expressing the subjective level of impact of an incident.	Single select
Incident effect	Expressing the possible effects of an incident.	Basic multiple selects
Security compromise	Expressing whether or not an incident resulted in a security compromise.	Single select
Loss Duration	Expressing the approximate length of time of a loss due to an incident.	Single select
Loss Property	Expressing the possible properties of a loss.	Basic multiple selects
Attacker Infrastructure	Expressing the type of infrastructure an attacker uses.	Basic multiple selects
Threat Actor Type	Expressing the type of a threat actor.	Basic multiple selects
Attacker Tool	Expressing types of attacker tools.	Basic multiple selects
Malware Type	Expressing types of malware instances.	Basic multiple selects
Malicious E-mail	Suspected malicious e-mail.	Multiple selects and user created items
IP Watchlist	A set of suspected malicious IP addresses or IP blocks.	Multiple selects and user created items
File Hash Watchlist	A set of hashes for suspected malicious files.	Multiple selects and user created items
Domain Watchlist	A set of suspected malicious domains.	Multiple selects and user created items
URL Watchlist	A set of suspected malicious URLS.	Multiple selects and user created items
Incident Category	Expressing the possible categories of an incident.	Single select
System Type	Expressing the type of a system.	Multiple selects and user created items
Asset Type	Expressing the type of an asset.	Multiple selects and user created items
Discovery Method	Expressing how an incident was discovered.	Multiple selects and user created items

Table 4.4: Shared items in incident information

4.3.4 User Profile

Different from DDoSGrid, the SHINE platform provides incident statistics and economic analysis for user and sector levels, so that a *User Profile* page is needed. The *User Profile* page provides user information management and modification functions and is also the entrance for adding countermeasures.

User Information

There are two kinds of user information, which are the basic information and the business profile. The basic information contains user's name, email address, country, city, postcode, and detailed address. And the business profile contains the business-related information, like business sector, company, user scale of the company, and the revenue of the company.

The system can perform statistics and analysis of data according to the sector or the organization, and calculate features and metrics from different incident and economic dimensions. Besides, the user scale and revenue data may become the possible features for the measure recommendation in our future works.

Cybersecurity Measures

The third part of the *User Profile* page is the cybersecurity measures information. The user could click the *Add Measure* button to add a new measure for a specific type of cyberattacks. And all of these added measures are listed in the *Cybersurity Measures* card.

When clicking the *Add Measure* button, an *Add a measure* dialog appears, similar with the *Information Sharing* dialog. For each kind of attack, the user could select different types of countermeasures that respond to these cyberattacks. To calculate the economic metrics of these countermeasures, several detailed data need to be filled, including the description of the measure, the active years of the measure, current discount rate, etc. The detailed information of these input items is listed in Table 4.5. After all of these data have been added, the user could click the *Save* button to save this measure into the database. Then the system will automatically compute economic metrics like ROSI and NPV for the added measure. And the user could view these economic metrics in *Userview*.

4.3.5 SHINE Views

As mentioned before, the main functions of the SHINE platform are information sharing, economic impacts analysis, incidents analysis, and data visualization. The information sharing function is embedded in the *Datasets* page, and the rest of the functions are mainly integrated in the SHINE views. The SHINE provides hierarchical views so that the user is able to get the analysis results of interested data from different depths. The *Overview* locates at the top level of our platform which gives a guideline for user's cybersecurity

situation. Below that are the *Sectorview* and the *Userview*, which analyze the data from a specific sector or organization respectively. At the bottom layer are the feature-specific views, which provide analysis from different dimensions.

Item name	**Item description**	**Input type**
Attack Type	The type of cyberattack that the user want to respond for.	Single select
Measure Type	The type of measure which the user want to use.	Single select
Measure Description	The description of this measure.	Text input
Measure Active Years	The description of this measure.	Text input
Vulnerability Without Measure	The possibility of being attacked by this type of cyberattack when there is no measures.	Text input
Discount Rate	Current discount rate.	Text input
Initial Cost	The initial cost of this measure.	Text input
Annual Upgrade Cost	The annual upgrade cost of this measure.	Text input
Annual Maintenance Cost	The annual maintenance cost of this measure.	Text input
Reduced Vulnerability	The possibility of being attacked by this type of cyberattack when the user uses this measures.	Text input
Reduced L1 Rate	The reduced L1 rate when the user uses this measures.	Text input
Reduced L2 Rate	The reduced L2 rate when the user uses this measures.	Text input

Table 4.5: Detailed information of the items in add a measure dialog

Overview

The *Overview* page is a snapshot of user's current situation of cybersecurity. There are two ways to reach this pare, either click the SHINE tab or click the *Overview* in the sidebar. Two types of statistical data are visualized on this page, which are incident statistics and economic impacts data. These charts provide basic views of how serious the consequences are caused by cyberattacks. To draw these charts, two types of parameters need to be passed into the front-end. One is the options of the chart, including whether grid lines are needed, etc., and the other is the data for the chart. And the data is provided by back-end applications through RESTful API. There are charts on this *Overview* page:

- A line chart shows the monthly losses caused by cyberattacks.
- A doughnut chart shows the share of loss caused by different types of cyberattacks.
- A stacked bar chart shows the monthly number of different types of cyberattacks.
- A polar area chart shows the total number of attacks in each attack type.

Sector View

The *Sectorview* page provides a way to analyze and visualize the cybersecurity status of the user from the perspective of the sector view. By default, this page analyzes and visualizes the data from the sector to which the user belongs, but our platform still provides a select area so that the user could also take a look at the results from another sector. Similar to the *Overview* page, the *Sectorview* page also analyzes and visualizes two aspects of data, including the incident statistics and the economic impacts. There are seven charts in this *Sectorview* page:

- A polar area chart portrays the total losses caused by cyberattacks in each sector.
- A bar chart shows the monthly losses of the chosen sector caused by cyberattacks.
- A doughnut chart shows the share of loss of the chosen sector caused by different types of cyberattacks.
- A stacked bar chart shows the monthly number of different types of cyberattacks in the chosen sector.
- A pie chart shows the total number of attacks in each attack type of the chosen sector.
- A bar chart shows the top 5 most vulnerable system in this chosen sector.
- A bar chart shows the top 5 most vulnerable asset in this chosen sector.

Feature Specific Views under Sectorview

If the user wants to view more specific results from different dimensions, he could go to the feature specific view pages to get the analyzed results. Six different feature specific sector view pages are provided in our platform, including the economic analysis, impact analysis, adversary analysis, attack analysis, asset analysis, and discovery method analysis. Not only can the user view these pages in his default sector, but also the other sector's feature specific view pages can be visited by changing the sector-select field. The features available as organized as follows.

- The economic analysis page provides four charts, including a polar area chart for the total losses in each sector, a bar chart for sector's monthly losses, a pie chart for sector's share of loss caused by different types of cyberattack, and a pie chart for sector's share of different types of loss.
- The impact analysis page provides five charts, including a stacked bar chart for the sector's monthly impact rating, a stacked bar chart for sector's loss duration statistics, a stacked bar chart for the sector's monthly security compromise, a stacked bar chart for sector's loss property statistics, and a bar chart for sector's impact effect statistics.
- The adversary analysis page provides four charts, including a stacked bar chart for sector's attacker tool statistics, a stacked bar chart for sector's malware type statistics, a stacked bar chart for sector's attacker infrastructure statistics, and a stacked bar chart for sector's threat actor type statistics.
- The attack analysis page provides six charts, including a stacked bar chart for sector's monthly incident category statistics, and bar charts for top IP, domain, URL, file hash, and malicious email watch list.
- The asset analysis page provides two charts, including a stacked bar chart for sector's attacked system type statistics, and a stacked bar chart for sector's attacked asset type statistics.
- The discovery method analysis page provides one chart, that is the bar chart for sector's discovery method statistics.

User View

Different from the *Sectorview* page, the *Userview* page provides a way to analyze and visualize the cybersecurity status of the user from the perspective of the organization view. And similar with the *Overview* and *Sectorview* page, the *Userview* page analyzes and visualizes from the incident and economic aspect. There are 7 charts on this *Userview* page:

- A bar chart shows the monthly losses of this organization caused by cyberattacks.
- A doughnut chart shows the share of loss of this organization caused by different types of cyberattacks.
- A polar area chart portrays the average losses caused by each type of cyberattacks in this organization.
- A stacked bar chart shows the monthly number of different types of cyberattacks in this organization.
- A pie chart shows the total number of attacks in each attack type of this organization.

- A bar chart shows the top 5 most vulnerable system in this organization.
- A bar chart shows the top 5 most vulnerable asset in this organization.

Feature Specific Views under User View

The same as sector view, the user view also provides several pages for feature specific views, including the economic analysis, impact analysis, adversary analysis, attack analysis, asset analysis, and discovery method analysis.

- The economic analysis page provides five charts, including a table for user's countermeasures analysis, a bar chart for user's monthly losses, a pie chart for user 's share of loss caused by different types of cyberattacks, a polar are chart for user's average loss in different types of cyberattacks, and a pie chart for sector's share of different types of loss.
- The impact analysis page provides five charts, including a stacked bar chart for the sector's monthly impact rating, a stacked bar chart for sector's loss duration statistics, a stacked bar chart for the sector's monthly security compromise, a stacked bar chart for sector's loss property statistics, and a bar chart for sector's impact effect statistics.
- The adversary analysis page provides four charts, including a stacked bar chart for sector's attacker tool statistics, a stacked bar chart for sector's malware type statistics, a stacked bar chart for sector's attacker infrastructure statistics, and a stacked bar chart for sector's threat actor type statistics.
- The attack analysis page provides six charts, including a stacked bar chart for sector's monthly incident category statistics, and bar charts for top IP, domain, URL, file hash, and malicious email watch list.
- The asset analysis page provides two charts, including a stacked bar chart for sector's attacked system type statistics, and a stacked bar chart for sector's attacked asset type statistics.
- The discovery method analysis page provides one chart, that is the bar chart for sector's discovery method statistics.

4.3.6 Component Structure

As aforementioned, our platform is a hierarchically structured system. The application consists of five tabs, *Home*, *Dashboard*, *Datasets*, *SHINE*, and *User*. This hierarchical structure of the components is shown in Figure 4.4. As we can see, the logic of our website starts with the *Landing page*, which is used to display the information of our project as well as the DDoSGrid project. The *Dashboard* tab and *Datasets* tab are inherited from DDoSGrid project. The *SHINE* tab is the main function of our platform, which provides

economic analysis and incident visualization features. The *User* tab is used as the entrance of user profile and countermeasure management modules.

Different depth of results analysis and visualization are available in the SHINE platform, including the *Overview*, the *Sectorview*, the *Userview*, and the *Feature Specific Views*. As the basis of our visualization, *Chart Components* could be treated as the infrastructure of our platform. The data is extracted from the back-end applications according to the defined logic and enters different chart modules. The front-end displays all the desired charts at the right places and in the right formats.

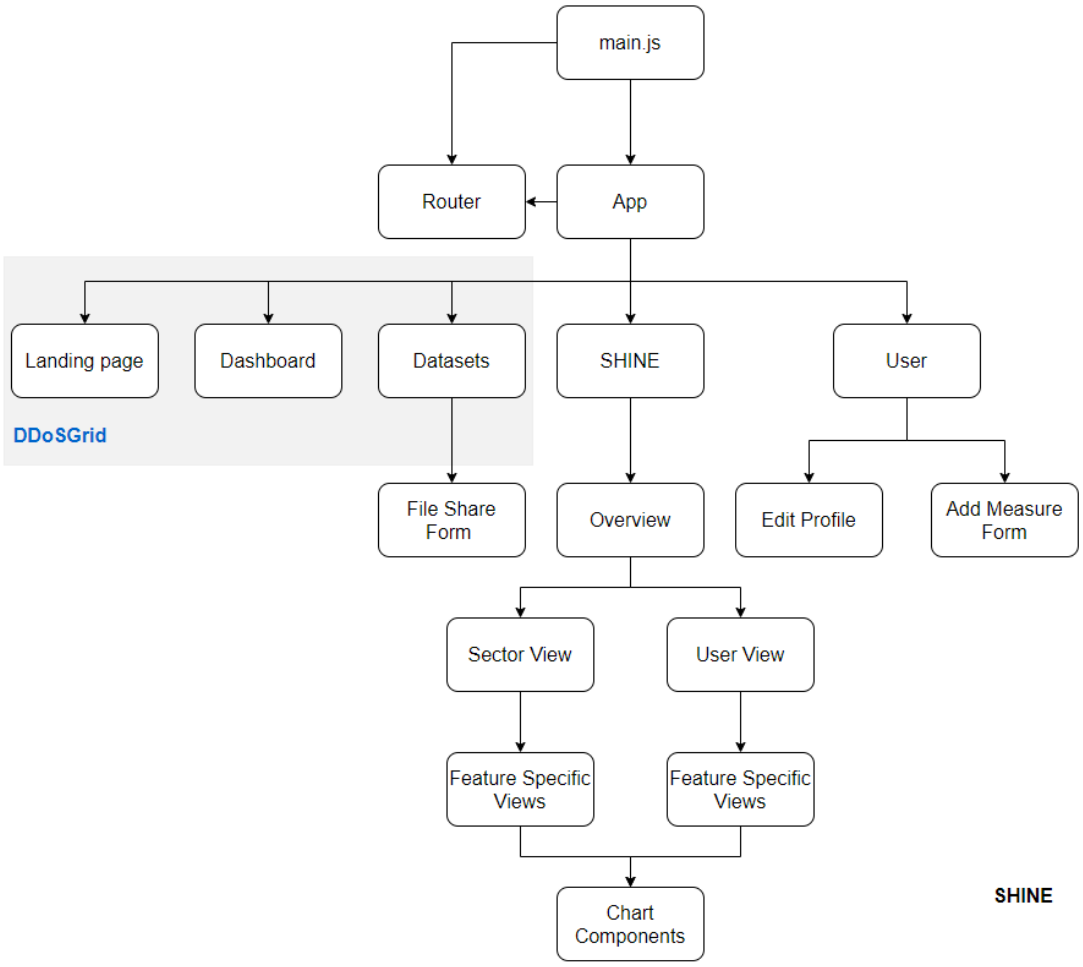

Figure 4.4: Front-end Component structure of the SHINE platform

4.3.7 Integration with DDoSGrid

As before mentioned, DDoSGrid is the baseline and start-point of our platform, some applications are inherited from the DDoSGrid. At the same time, to adapt to the SHINE system, we have also modified some applications of the DDoSGrid system. In the subsection, we are going to discuss how to integrate out system with the previous system.

DDoSGrid is mainly contains three sub-projects, including the *Miner*, the *API*, and the *Frontend*. *Miner* and *API* are the backend, data-bus, and information processing center

for their system. And we have not made any change to these parts, but directly inherited their codes and functions. This also means that the backend of our system and their backend are isolated to each other.

On the frontend side, as mentioned earlier, we have changed such as the *Datasets* page, *Landing* page, etc., so we need to directly cover these relevant pages of the previous system. At the same time, our newly added page only needs to be pasted to the corresponding position. In this way, our new functions and modules on these pages can be integrated into the previous system.

In terms of authentication, we directly follow the method of the DDoSGrid system, that is, use the DDoSDB system and Oauth2 module for authentication. Since there are two different user management database tables in these two systems, there will be two different user states: A. the user exists in the DDoSDB system, but does not exist in the SHINE system; B. the user exists in both systems at the same time. The latter state is what we want, but for the former, we also have dealt with it accordingly. To get through the user management modules of the two systems, if the user clicks the *SHINE* tab or the *User* tab, the SHINE system will automatically obtain the user's information from the DDoSDB system, and then add a row of user records to the *Userinfo* table in the SHINE system. As a result, users will exist in both systems, and can use the functions of the two systems at the same time

Chapter 5

Evaluation

5.1 Case Studies

In order to assess the relative features of the platform, three case studies for stakeholders with different purposes are executed in the following subsections. The first case considers a computer security expert while the rest of two regard a cybersecurity expert and also decision-maker of an organization as the user. Each of the case studies follows certain procedures until the objects of the users are fulfilled.

5.1.1 Case Study #1: Sharing Information with Interested Stakeholders

In the first scenario, we assume the user is a computer security expert that would like to improve the cybersecurity for the companies in the whole sector. Furthermore, we suppose that the user wants to provide as much information concerning cybersecurity in a specific sector as possible so that other users on the platform could acquire this information and then have a reference for their own business. Therefore, the information sharing function is chiefly used in this scenario after the user has signed in.

The log-in procedures are similar for each user, that after signing up and having an account, the user signs in the system through the interface on the Home page, input the user name and password, and gain authorization from the DDoSDB system.

The uploading process happens on the *Dataset* tab. For each attack, the user needs to first upload the network traffic logs through the *UPLOAD A DATASET* button on the center or the *Upload a raw PCAP file* button on the right corner. Followed by a new card is created on the file, the name of the dataset and the description are added. The name, id, uploading time, the status of feature extraction, file size, and analysis duration is then displayed upon the file card, which is similar to that on the DDoSGrid. Similarly, as long as the files have been uploaded, several technique features, for instance, source IP, are then mined and processed automatically.

Besides the log files, for the purpose of sharing other attack-related information, the button *SHARE* on the bottom of the card is pressed, and a window pops out immediately for the user to input this information, including basic attack information involving the sector and organization in *Basic Information* tab, economic-irrelevant impacts of the attack in *Information Sharing* tab, and economic-related impacts of the attack in *Economic Impacts* tab.

The *Economic Impacts* section asks the user to feed in economic-related impacts due to a specific attack, for instance, the cost of equipment replacement, cost of repair, corporate income loss, loss due to non-compliance with the contract, and other indirect loss. And the *Information Sharing* section concerns other impacts caused by the attack, including several parameters in the combo box, such as impact rating, incident effect, loss property, attacker infrastructure, threat actor type, attacker tool, malware type, attack type, system type, and asset type, as well as other parameters, such as security compromise, malicious e-mail, IP watchlist, file hash watchlist, domain watchlist, and URL watchlist.

After the impact information mentioned above is filled, the user can confirm the information and push it into the system by clicking the *SAVE* button. The data included is then available to all the users on the platform. Nevertheless, the user could still clear the dataset and all the information related to it by the *DELETE* button on the file card. The *Share More Information* is shown as Figure 5.1. Finally, the user repeats the uploading procedure above until all the data is shared.

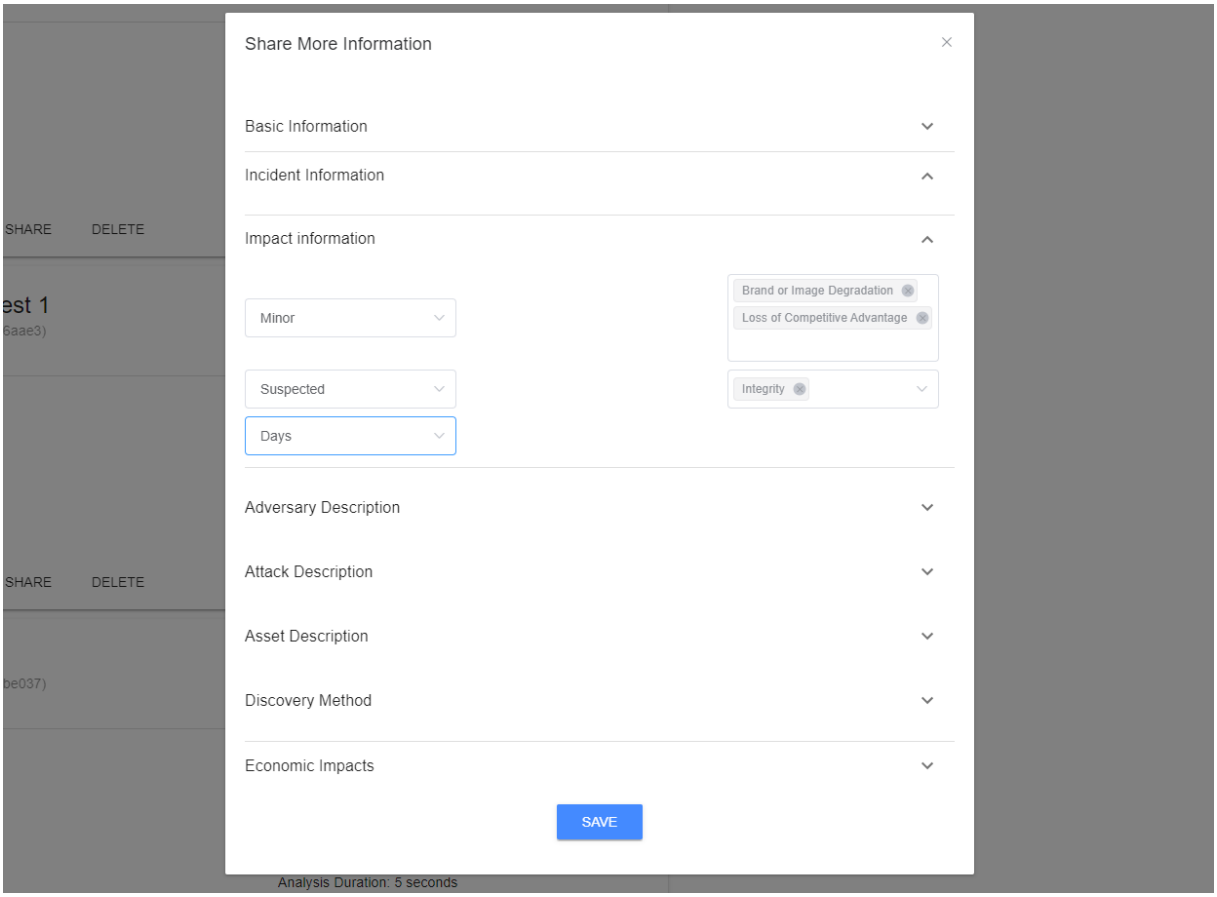

Figure 5.1: Attack Information Sharing

5.1.2 Case Study #2: Analysis of Threats in a Specific Sector

In the second scenario, a Cybersecurity expert as well as decision-maker of an organization in a specific sector, for example, the finance sector is considered. For this scenario, we suppose that the user wants to review the overall situation about the cyberattacks in the last year so that he could find a reference for which categories of cybersecurity to pay attention to or invest in for the next year. Besides, we assume that several relative datasets and sufficient information have been uploaded and shared by other users, and are available to this user after log-in. For the reason that the more information sets are included, the more reasonable and comprehensive the results are, the user is supposed to not only acquire information from others' shared datasets but also upload his own datasets. Basically, the signing in and uploading process are similar to the first case.

When the pre-processing phase is complete, the user jumps to the *SHINE* tabs to find the final results there. On the *SHINE* page, the most critical results of economic impact and incident statistics regarding the organization are displayed firstly on default with two charts in each section. The following charts are on show in Figure 5.2:

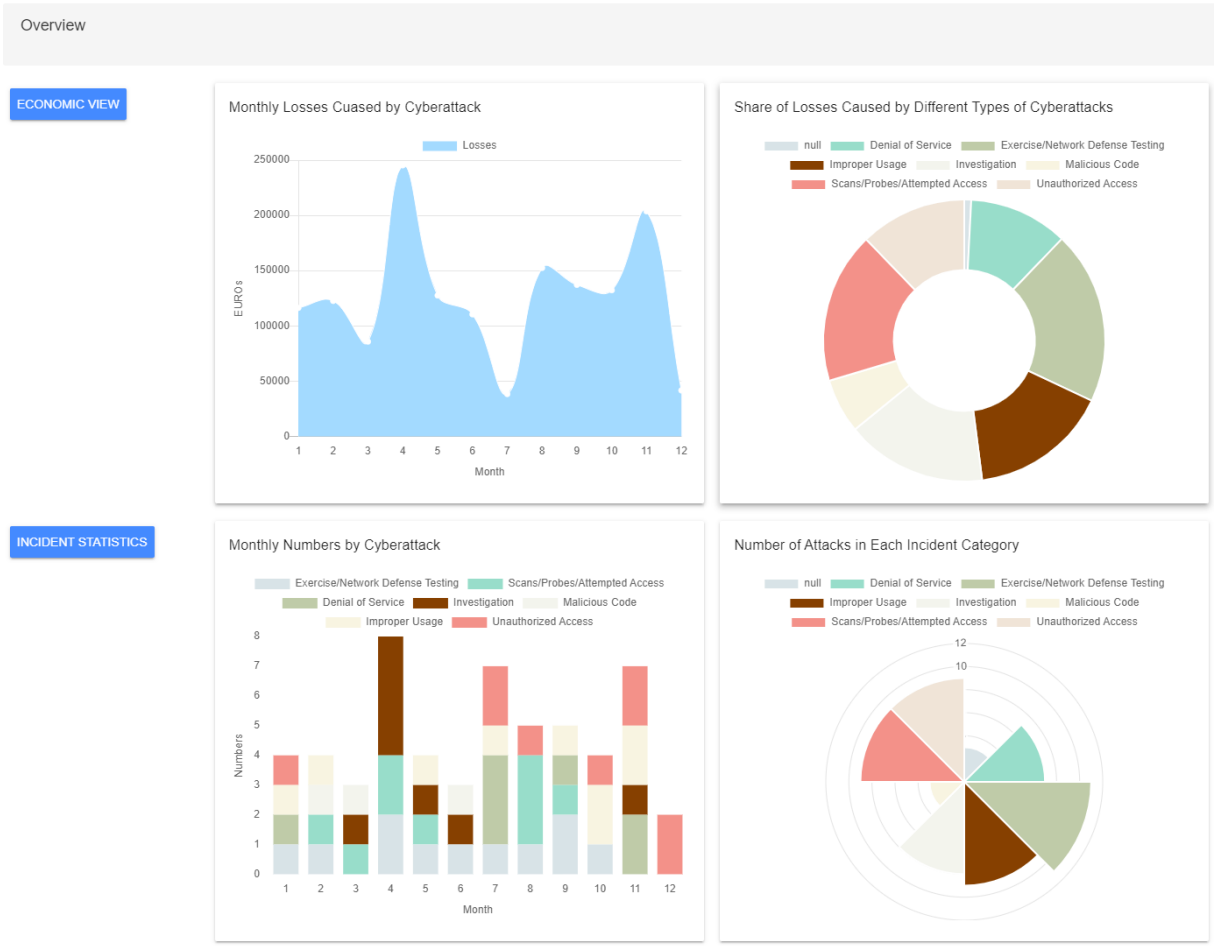

Figure 5.2: Overview page in SHINE tab

The first one in the economic impacts section is a time distributing chart concerning monthly losses caused by cyberattacks. The user could obtain some insights from this graph, for instance, which month of a year is more vulnerable under all types of cyberattacks. The second one is a pie chart showing the proportion of losses caused by various cyberattacks in a certain period for all sectors. From this graph, the user could find the cyberattacks that lead to the most severe financial aftermath for his business.

For the incident statistics section, the first one is a stacked bar chart reflecting monthly numbers of cyberattacks incident occurring in an organization. Each of the bars shows the number of attacks in each month, while each of the colors composing the bar represents the number of a type of attacks. For example, the gray part in the first bar means that the Exercise/Network Defense Testing in January appears once. This chart could uncover either the composition of attacks appearing in each month or the overall number of attacks across the whole year.

The second chart for the incident statistics section is a radar chart demonstrating the number of attacks in each incident category for an organization in a year. Each color of a sector represents an incident, and a larger space of a color reveals the higher occurrence probability of a cyberattack incident.

Since the user in this scenario also requests for a depth of sector-specific level, thus, the *More* button for sector-specific view either on the Economic impacts or on the Incident statistics section is pressed and then it is redirected to the sector view page. In the *Sector view* page, after selection of *Finance* in the sector box, the charts for different impacts of cyberattacks are shown on the right. To enable the user to find the features easier, a menu on the left of the section name is offered to select particular metrics. Similarly, page of charts for the sector could be found as follow:

In the Economic View section, which is shown in Figure 5.3, the radar chart exhibiting the total losses in each sector listed at first. Each block represents a sector of business, for example, the gray block serves as the finance sector, while the red means the educational sector. The larger the block is, the bigger the losses generated in this sector are. Generally, it compares the overall losses in each sector.

Similar to the economic impacts drawn on the *Overview* page, the second Figure concerns the monthly losses in the whole sector caused by cyberattacks. Consequently, it reminds the users that which months might have a relatively higher possibility to be attacked in an exact sector. The third one in the Economic view is a pie chart reflecting the percentage of losses caused by various types of attacks in a certain time period is given. The user could find intuitively the attacks that result in more financial losses for the whole sector.

In the Incident Statistics section, which is shown in Figure 5.4, the stacked bar chart named Monthly Number of Attacks is in the first place. Analogously, the frequency of each type of attacks in each month is piled, thus the results could be reviewed by vertical in each month or horizontal across different months. The second chart in the Incident Statistics section is a pie chart showing how many attacks happened in the whole sector for each of the incident categories, which is represented by a certain sector with a unique color.

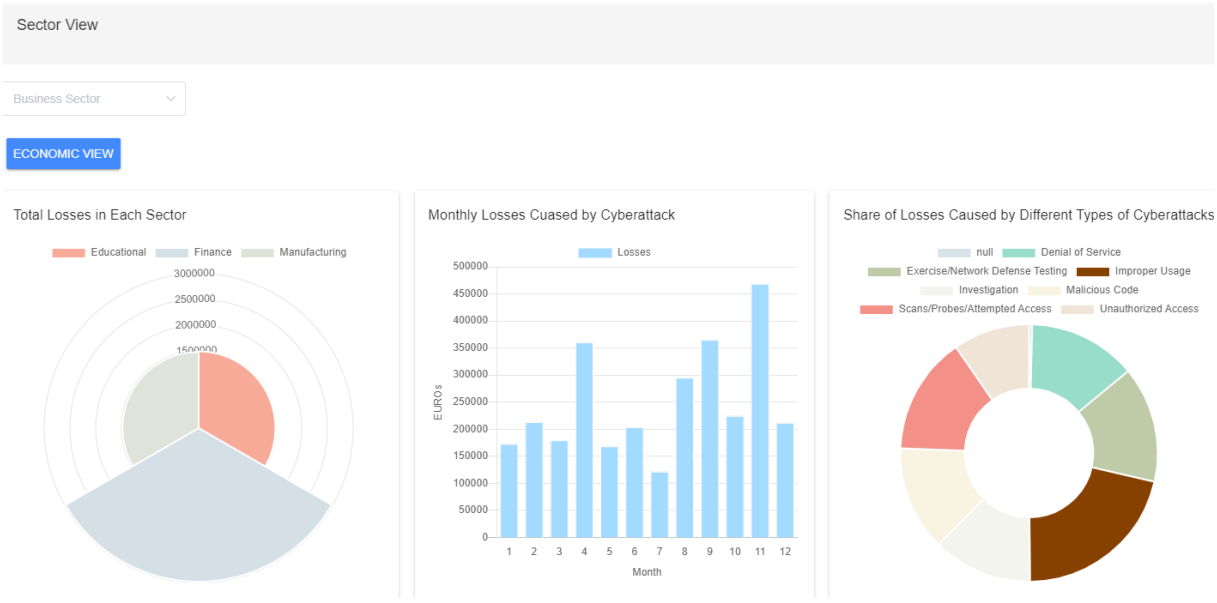

Figure 5.3: Sector view page in SHINE tab (part 1)

Figure 5.4: Sector view page in SHINE tab (part 2)

The last two graphs in this section are bar charts demonstrating the top 5 attacked systems and attacked assets respectively. With showing the attacked frequency on the y-axis, the graphs listed the most common attacked systems (*e.g.* Enterprise Systems, Third-Party Services and *et al.*) and assets (*e.g.* Administrator and Database). Insights could be gained by synchronizing some of these figures. For instance, the most common type of cyberattack for the user, in this case, is Network defense testing, while for the whole sector, that is improper usage. Therefore, the user might need to pay more attention to both of the types to reduce the risks of being attacked.

As over ten features are contained in the sector view as well as the user view, to enable the user to have deep analysis results of the whole sector and find the features easier, a menu on the left of the section name is offered to select a particular category of metrics (*i.e.* Impact Analysis, Adversary Analysis, Attack Analysis, Asset Analysis, and Discovery Method Analysis). For example, the subsection *Impact Analysis* in the *Incident Statistics* section consists of the results of impact rating, loss duration, security compromise loss property, and incident effect, where the user could have easy access to them by clicking the *Impact Analysis* tab on the left menu. And if the user wonders which layer of the system is most vulnerable, The scene above is showed in Figure 5.5.

Figure 5.5: Incident Statistics results in Sector view page

5.1.3 Case Study #3: Insights of Threats and Investments for a Business

In the last scenario, we suppose it has the same user and is based on similar assumptions as in the second case. Besides, we assume that the user has experienced the second case, which means the login and uploading procedures are completed, and the user has known the overall situation about the cyberattacks in the last year. The difference is that the user in this case would mainly focus on gaining more insights about attacks against his own business, especially for the countermeasures against attacks, which requests a user-specific level of results visualization. As cybersecurity strategies are commonly regarded to be private for a company, in this case, we presume that all the relative datasets and other information are uploaded by the user.

By clicking the *User view* term under *SHINE* tab, the user switches to the user-specific view page and has access to the detailed related results. For example, we could find the monthly losses caused by cyberattacks, which is similar to that on the overview page, and the average losses caused by each type of attack under user view. It is unexpected that though DDoS is listed as the forth-common type of attack, it generates the largest average loss. The user then decides to have a further investigation on the DDoS attack, especially the economic impacts and countermeasures. To research the economic aspects, the measures about how to combat a specific type of attack are requested from the user at an interface in *User* page as shown in Figure 5.6. Followed by popping out of the window *Measure Details*, the user presses the *Add Measure* button at the bottom of the page. The basic information including *Attack Type* and *Measure Type* are then set to be Denial of Service and Periodically security awareness training respectively.

After inputting the brief description of the measure, several parameters concerning the measures are required. *Measure Active Years*, the year about how long the measure lasts, settle to four years. And the initial cost and annual upgrade and maintenance cost are 100000, 100 and 100 respectively. Other parameters regarding the efficiency of the measure, such as *Vulnerability Without Measure*, *Reduced Vulnerability*, *Reduced L1 Rate*, *Reduced L2 Rate* and *Discount Rate* have default values, predefined by the system according to the experience of the experts and relative literature.

The *Vulnerability Without Measure* is a percentage measuring the possibility that the organization is attacked successfully by a type of attack. And the *Reduced Vulnerability*, *Reduced L1 Rate*, and *Reduced L2 Rate*e represent the efficiency rates of the countermeasure, reducing the overall probability of being attacked, the losses generated during repairing the attack, and the losses generated during detecting the attack. The *Discount Rate* is an interest rate that determines the present value of the future cash flows, which is mostly set from 0.05 to 0.10.

As the measures are added into the database and displayed on the *User* page, the user turns back to the *User view* page under the *SHINE* tab. In the *User Economic Impact Analysis* section, the economic metrics (*i.e.* ROSI and NPV) of the measure is then calculated and displayed as displayed in Figure 5.7. The ROSI (*i.e.* Return on Security Investment) assesses the price-performance ratio of the security measure. Thus, a positive value means a valuable investment and the measure with a higher value of ROSI generally

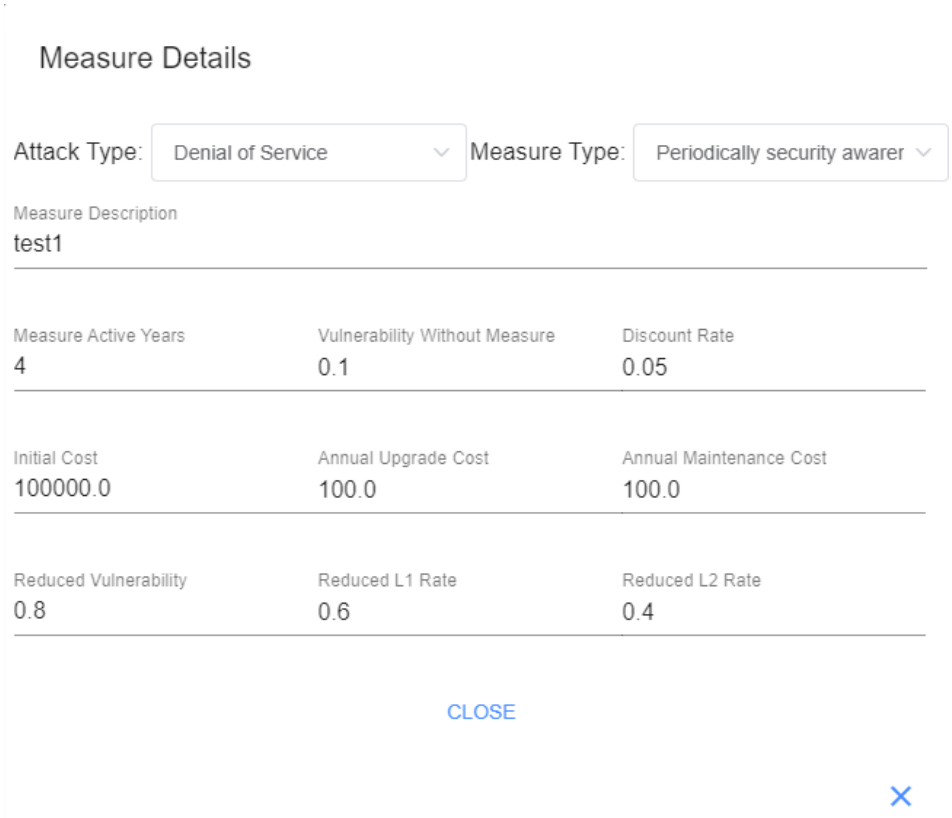

Figure 5.6: Measure Details updating page

represents it is more worthy to invest. The NPV (*i.e.* Net Present Value) reflects the present value of future cash flows, usually applied to compare different measures under the same time period. Similarly, the measure with a higher value of NPV also represents it is more worthy to invest. In consequence, the user could compare these self-defined countermeasures by the ROSI and NPV values, and decide which one to invest to combat a particular type of attack.

5.1.4 Discussion

From the studies above, the majority of the features of the SHINE platform are covered during the cases, and the stakeholders with different purposes could benefit from the system. For example, for the cybersecurity experts who would like to improve the security situation for the whole industry, SHINE enables them to share essential information with others.

As for those who want to gain insights on cyberattacks especially the economic impacts through the SHINE system, the platform allows them to collect, process, and calculate the relative uploaded attack information, and gives out intuitive tables and visualization results. The insights are organized in several levels and aspects. For instance, the user can check the most critical features and results on the Overview page, such as which types of attack occur the most, or which types of attack generate the largest losses. To have an inspection on the overall conditions of a specific sector, the user could switch to the Sector

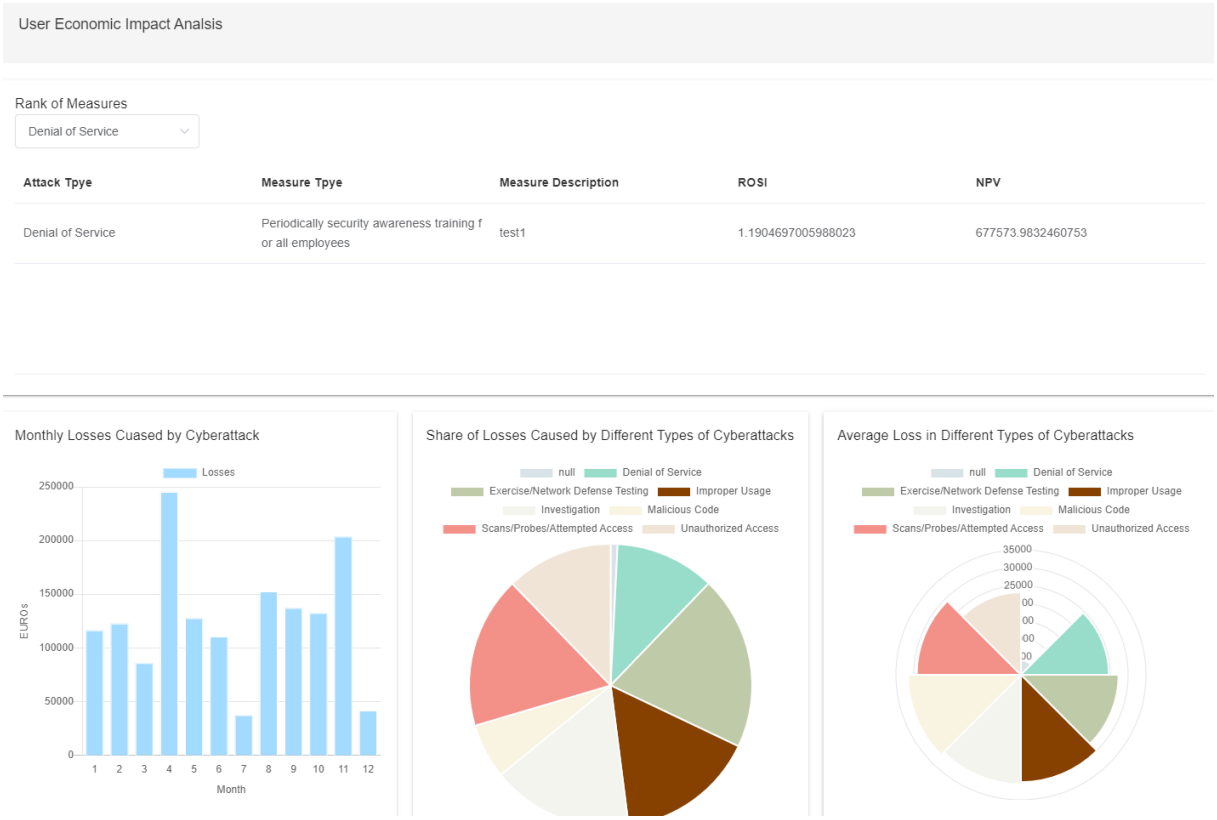

Figure 5.7: User overview results in SHINE tab

view and find relative figures. And in the User view page, the user could first acquire the economic-related information, including the ROSI and NPV, after the countermeasures against a specific category of attacks being added to the User page. These economic metrics indicate which protective measures have the highest value to invest, and could support the user to decide which and how to invest during cybersecurity strategy decision making process. Furthermore, the results about economic impacts, such as the comparison of average losses among incident categories, as well as the results about incident statistics, for instance, the duration of the attacks are displayed in both the *Sector view* and the *User view*.

Therefore, the user could build up a deeper understanding about the impacts and strategies of the cyberattacks through the SHINE system from various perspectives, and then allocate reasonable resources to moderate or eliminate the impacts based on the information. Basically, the SHINE platform could serve as an efficient tool for relative stakeholders to achieve their goals on information sharing and insights gaining of cyberattacks.

However, the SHINE system still has limitations and needs to improve to some extent. First of all, the economic model and metrics applied now are ideal and simplified. It rests on the assumption that the user has the knowledge of the details of the economic impacts and countermeasures. For example, the user would have to first input different losses into the system through the information sharing interface, which means, for each type of the losses (*i.e.* cost of equipment replacement, cost of repair, corporate income loss, organization productive loss, loss due non-compliance with statutory provisions and

indirect loss) generated during each phase of attacks (*i.e.* detection time and repair time). Then the calculation of the economic metrics, ROSI and NPV, requests the user to fill in the countermeasure form in the *User* page. Among it, several parameters (*e.g.* vulnerability without measures) are calculated and put in by the user. Nevertheless, the other parameters (*i.e.* reduced vulnerability, reduced L1 rate, reduced L2 rate, discount rate) are pre-defined by the system according to the experience of the experts, though the user could also edit the pre-defined number.

As stated in the original model we referred to, the losses are categorized in three parts (*i.e.* fixed losses, losses during detecting the attack, and losses during repairing the attack) based on in which phase they are generated [29]. And this model assumes that the countermeasures also have three categories (*i.e.* measures reducing the vulnerability, measures reducing the detecting time, measures reducing the repairing time). To simplify the calculation, detecting time and repairing time in the original model are replaced by the reduced rate of losses in detecting time and repairing time in our model. This leads to a disadvantage that the users need to calculate these rates by themselves, otherwise they will have to accept the default rates defined by the system. And as these rates are estimates of the measures, inaccurate estimation might somewhat influence the final results of the ROSI and NPV, thus makes the comparison to be not sufficiently precise. The suggestions for how to overcome the limitation will be covered in the future work in the next chapter.

Chapter 6

Summary, Conclusions and Future Work

This chapter is about to summarize the entire project and bring up some future possibilities. In the first section, we are going to summarizes what we did to accomplish this project and then draw some conclusions about the SHINE system. The expectation of the future work with a brief explanation will be described in the second section.

6.1 Summary and Conclusion

In order to achieve the goal in the project proposal, which is to design and implement a collaborative platform that allows different stakeholders to share information and gain insights from analysis of the cyberattack information and economic impacts of attack, following efforts and work are made.

Firstly, we extracted the needs of developing a such platform and investigated the background knowledge and related works. After the requirements extraction process, we sketched the architecture of the SHINE system and a high-level mock-up. This step allowed us to define the structure, design the functionality and research the feasibility of the system. Following, we implemented the SHINE system including the implementation of the defined data models, front-end and back-end, as well as integration with the DDoS-Grid system. Lastly, the SHINE system is evaluated and tested by applying it under several scenarios with different roles of users.

From the user's perspective, the implemented SHINE system could cover the main goal and work properly for different users with various purposes. The SHINE platform, integrating the functions such as information sharing, attack information processing, economic metrics calculating, and results visualizing, could help stakeholders from different sectors to comprehend the financial and technical impacts of the attacks and make more plausible decisions on cybersecurity.

6.2 Future Work

At first, our work on information sharing is basic, which means at the current stage that the user could share the data with all other users on the platform. Allowing the user to select and decide which user or which group to share might be a nice extension in the future. Next, for the analysis of the economic impact, the evaluation of investment on the countermeasures facing a dilemma that the models applied currently are still not precise and accurate, and are based on multiple assumptions and not perfectly practical. Since only the ROSI and NPV metrics are used in the current phase, for the following work, involving more updated quantitative models even the qualitative models to assess the economic impacts from different aspects would improve the performance of the platform undoubtedly.

Furthermore, as the measures against cyberattacks are pre-defined in the system, to have cybersecurity service recommendations embedded in SHINE could extend the functions to be more practical. In 2019, Muriel *et.al.* developed the MENTOR system, which could recommend off-site protection services for users to address specific attacks [21]. Hence for the next step, integrating with the MENTOR system and reorganizing the recommendation of cybersecurity measures in SHINE is expected. Finally, if the volume of the sharing data on the platform continues to rise, then it is foreseeable to enable the users to gain suggestions for their cybersecurity investment based on other similar users. And Artificial Intelligence algorithms like machine learning or deep learning might have advantages in processing a high volume of data under such a situation.

Bibliography

[1] Abhishta Abhishta. *The blind man and the elephant: Measuring economic impacts of ddos attacks*. University of Twente, 2019.

[2] Ross Anderson et al. "Measuring the cost of cybercrime". In: *The economics of information security and privacy*. Springer, 2013, pp. 265–300.

[3] Jan von der Assen. *DDoSGrid 2.0: Integrating and Providing Visualizations for the European DDoS Clearing House*. 2021.

[4] David Balson and William Dixon. "Cyber information sharing: building collective security". In: World Economic Forum.

[5] Sean Barnum. "Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX)". In: *Mitre Corporation* 11 (2012), pp. 1–22.

[6] David Bianco. *The Pyramid of Pain*. URL: `https://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html`.

[7] K Bissell, R LaSalle, and PD Cin. "Ninth annual cost of cybercrime study". In: *Ponemon Institute: Dublin, Ireland* 6 (2019).

[8] Lawrence D Bodin, Lawrence A Gordon, and Martin P Loeb. "Evaluating information security investments using the analytic hierarchy process". In: *Communications of the ACM* 48.2 (2005), pp. 78–83.

[9] Luc Boillat. "DDoSGrid-Mining: Analyzing and Classifying DDoS Attack Traffic". In: *Communication Systems Group, Department of Informatics, Universität Zürich* (2021).

[10] Luc Boillat et al. *A Tool for Visualization and Analysis of Distributed Denial-of-Service (DDoS) Attacks*. 2020.

[11] *CiviCERT – The Computer Incident Response Center for Civil Society*. en-US. URL: `https://www.civicert.org/` (visited on 03/11/2021).

[12] Marco Cremonini and Patrizia Martini. "Evaluating information security investments from attackers perspective: the return-on-attack (ROA)". In: *WEIS*. 2005.

[13] *Cyber Threat Alliance (CTA) - Home*. en-US. URL: `https://www.cyberthreatalliance.org/` (visited on 03/11/2021).

[14] *DDoS Clearing House: "DDoSDB"*. en-US. URL: `https://ddosdb.org/about` (visited on 09/11/2020).

[15] *ddos-clearing-house/ddosdb*. en. URL: https://github.com/ddos-clearing-house/ddosdb (visited on 03/16/2021).

[16] Lukas Demetz and Daniel Bachlechner. "To invest or not to invest? Assessing the economic viability of a policy and security configuration management tool". In: *The economics of information security and privacy*. Springer, 2013, pp. 25–47.

[17] *Element UI*. en-US. URL: https://element.eleme.io/#/en-US (visited on 04/19/2021).

[18] Greg Farnham and Kees Leune. "Tools and standards for cyber threat intelligence projects". In: *SANS Institute* (2013).

[19] *Forecast Analysis: Information Security, Worldwide, 2Q18 Update*. en. URL: https://www.gartner.com/en/documents/3889055/forecast-analysis-information-security-worldwide-2q18-up (visited on 03/16/2021).

[20] M. Franco et al. "DDoSGrid: A Platform for the Post-Mortem Analysis and Visualization of DDoS Attacks". In: *20th IFIP Networking (Networking 2021), Posters and Demo*. Espoo, Finland: IFIP, June 2021, pp. 1–3.

[21] M. F. Franco, B. Rodrigues, and B. Stiller. "MENTOR: The Design and Evaluation of a Protection Services Recommender System". In: *15th International Conference on Network and Service Management (CNSM 2019)*. Halifax, Canada, Oct. 2019, pp. 1–7.

[22] Frank Fransen, Andre Smulders, and Richard Kerkdijk. "Cyber security information exchange to gain insight into the effects of cyber threats and incidents". In: *e & i Elektrotechnik und Informationstechnik* 132.2 (2015), pp. 106–112.

[23] Gustavo Gonzalez-Granadillo et al. "Enhancing Information Sharing and Visualization Capabilities in Security Data Analytic Platforms". In: *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE. 2019, pp. 1–8.

[24] Cristin Goodwin et al. "A framework for cybersecurity information sharing and risk reduction". In: *Microsoft* (2015).

[25] Lawrence A Gordon and Martin P Loeb. "The economics of information security investment". In: *ACM Transactions on Information and System Security (TISSEC)* 5.4 (2002), pp. 438–457.

[26] *Home - Django REST framework*. URL: https://www.django-rest-framework.org/ (visited on 04/20/2021).

[27] Muhammad Al-Humaigani and DerrekB Dunn. "A model of return on investment for information systems security". In: *2003 46th Midwest Symposium on Circuits and Systems*. Vol. 1. IEEE. 2003, pp. 483–485.

[28] Borka Jerman-Blažič et al. "An economic modelling approach to information security risk management". In: *International Journal of Information Management* 28.5 (2008), pp. 413–422.

[29] Borka Jerman-Blažič et al. "Quantitative model for economic analyses of information security investment in an enterprise information system". In: *Organizacija* 45.6 (2012), pp. 276–288.

[30] Christopher Johnson et al. *Guide to cyber threat information sharing*. Tech. rep. National Institute of Standards and Technology, 2016.

[31] Kimberly Lukin. "Russian Cyberwarfare Taxonomy and Cybersecurity Contradictions Between Russia and EU: An Analysis of Management, Strategies, Standards, and Legal Aspects". In: *National Security: Breakthroughs in Research and Practice*. IGI Global, 2019, pp. 408–425.

[32] Kanta Matsuura. "Productivity space of information security in an extension of the Gordon-Loeb's InvestmentModel". In: *Managing Information Risk and the Economics of security*. Springer, 2009, pp. 99–119.

[33] Ali Pala and Jun Zhuang. "Information sharing in cybersecurity: A review". In: *Decision Analysis* 16.3 (2019), pp. 172–196.

[34] P Pawlinski et al. "Actionable Information for Security Incident Response". In: *European Union Agency for Network and Information Security, Heraklion, Greece* (2014).

[35] *Press Release*. URL: `https://www.mcafee.com/enterprise/en-us/about/newsroom/press-releases/press-release.html?news_id=6859bd8c-9304-4147-bdab-32b35457e629` (visited on 05/01/2021).

[36] Real Python. *Django Rest Framework – An Introduction – Real Python*. en. URL: `https://realpython.com/django-rest-framework-quick-start/` (visited on 04/20/2021).

[37] Loren Paul Rees et al. "Decision support for Cybersecurity risk planning". In: *Decision Support Systems* 51.3 (2011), pp. 493–505.

[38] B. Rodrigues et al. "SEConomy: A Framework for the Economic Assessment of Cybersecurity ". In: *16th International Conference on the Economics of Grids, Clouds, Systems, and Services (GECON 2019)*. Leeds, UK: Springer LNCS, Sept. 2019, pp. 1–9.

[39] *RVAs3c: David Bianco: Pyramid of Pain: Intel-Driven Detection/Response to Increase Adversary's Cost - YouTube*. URL: `https://www.youtube.com/watch?v=zlAWbdSlhaQ` (visited on 01/14/2021).

[40] *Schema documentation for*. URL: `https://stix.mitre.org/language/version1.0/xsddocs/default_vocabularies/1.0.0/stix_default_vocabularies.html` (visited on 04/25/2021).

[41] Wes Sonnenreich, Jason Albanese, Bruce Stout, et al. "Return on security investment (ROSI)-a practical quantitative model". In: *Journal of Research and practice in Information Technology* 38.1 (2006), p. 45.

[42] Linda J Tallau, Manish Gupta, and Raj Sharman. "Information security investment decisions: evaluating the Balanced Scorecard method". In: *International Journal of Business Information Systems* 5.1 (2010), pp. 34–57.

[43] *The Django admin site | Django documentation | Django*. URL: `https://docs.djangoproject.com/en/3.1/ref/contrib/admin/` (visited on 04/01/2021).

[44] *Vue Material*. en-US. URL: `https://vuematerial.io/components/app` (visited on 04/19/2021).

[45] *vue-chartjs*. en-US. URL: `https://vue-chartjs.org/` (visited on 04/19/2021).

[46] *vue-sidebar-menu*. en-US. URL: https://yaminncco.github.io/vue-sidebar-menu/#/ (visited on 04/19/2021).

[47] *Vue.js The Progressive JavaScript Framework*. en-US. URL: https://vuejs.org/index.html (visited on 04/19/2021).

[48] Jan Willemson. "Extending the Gordon and Loeb model for information security investment". In: *2010 International Conference on Availability, Reliability and Security*. IEEE. 2010, pp. 258–261.

[49] Denise E Zheng and James A Lewis. "Cyber threat information sharing". In: *Center for Strategic and International Studies* (2015).

Abbreviations

AHP	Analytic hierarchy process
API	Application Programming Interface
CISA	Cybersecurity Information Sharing Act
CiviCERT	A network of Computer Emergency Response Teams
CSG	Communication Systems Group
CSS	Cascading Style Sheets
CTA	The Cyber Threat Alliance
DDoS	Distributed Denial-of-Service
E-R model	Entity–Relationship Model
FS-ISAC	The Financial Services Information Sharing and Analysis Center
HTML	The HyperText Markup Language
IPC	Interprocess Communication
IP	The Internet Protocol
IRR	Internal Rate of Return
IT	Information Technology
JSON	JavaScript Object Notation
l1	Losses generated during repairing time
l2	Losses generated during detecting time
l3	Fixed losses
NPV	Net Present Value
MM-ISAC	The Mining and Metals Information Sharing and Analysis Center
NPV	Net Present Value
PCAP	Packet capture
ROA	Return-on-Attack
R0	Risk without taking measures
RC	Risk after taking measures
REST	Representational state transfer
ROI	Return on Investments
ROSI	Return on Security Investment
SQL	Structured Query Language
STIX	Structured Threat Information Expression
T-ISAC	Telecommunication Information Sharing and Analysis Center
TTP	Tactics, Techniques and Procedures
USD	The United States dollar
URL	Uniform Resource Locator
URLConf	URL configuration

UZH	Universityof Zurich
VaR	Value-at Risk

List of Figures

List of Tables

Appendix A

Enumerate field content

This appendix illustrates the items in each enumerate field of our database. Developers and administrators can modify, add and delete options in the fields according to their needs. Table A.1 lists all the items in these enumerate fields.

Enumerate field	**Items**
Asset Type	Assettype
	Attackerinfrastructure
	Attackertool
	Impactrating
	Incidentcategory
	Incidenteffect
	Lossproperty
	Malwaretype
	Measuretype
	Securitycompromise
	Systemtype
	Threatactortype
	Discoverymethod
	Backup
	Database
	Mail
	Server
	Network
	Telephone
	Administrator
	Customer
	Employee
	Finance
	Hardware
	Partner
	Unknown

Attacker infrastructure	Anonymization Anonymization - Proxy Anonymization - TOR Network Anonymization - VPN Communications Communications - Blogs Communications - Forums Communications - Internet Relay Chat Communications - Micro-Blogs Communications - Mobile Communications Communications - Social Networks Communications - User-Generated Content Websites Domain Registration Domain Registration - Dynamic DNS Services Domain Registration - Legitimate Domain Registration Services Domain Registration - Malicious Domain Registrars Domain Registration - Top-Level Domain Registrars Hosting Hosting - Bulletproof / Rogue Hosting Hosting - Cloud Hosting Hosting - Compromised Server Hosting - Fast Flux Botnet Hosting Hosting - Legitimate Hosting Electronic Payment Methods
Attacker Tool	Malware Penetration Testing Port Scanner Traffic Scanner Vulnerability Scanner Application Scanner Password Cracking
Discovery Method	Agent Disclosure Fraud Detection Monitoring Service Law Enforcement Customer Unrelated Party Audit Antivirus Incident Response Financial Audit Fraud Detection HIPS IT Audit

	Log Review NIDS Security Alarm User Unknown
Impact Rating	None Minor Moderate Major Unknown
Incident Effect	Brand or Image Degradation Loss of Competitive Advantage Data Breach or Compromise Degradation of Service Destruction Disruption of Service / Operations Financial Loss Loss of Confidential / Proprietary Information or Intellectual Property Regulatory, Compliance or Legal Impact Unintended Access User Data Loss
Incident Category	Exercise/Network Defense Testing Unauthorized Access Denial of Service Malicious Code Improper Usage Scans/Probes/Attempted Access Investigation
Loss Duration	Permanent Weeks Days Hours Minutes Seconds Unknown
Loss Property	Confidentiality Integrity Availability Accountability Non-Repudiation
Malware Type	Automated Transfer Scripts Adware Dialer Bot Bot - Credential Theft Bot - DDoS

	Bot - Loader Bot - Spam DoS / DDoS DoS / DDoS - Participatory DoS / DDoS - Script DoS / DDoS - Stress Test Tools Exploit Kits POS / ATM Malware Ransomware Remote Access Trojan Rogue Antivirus Rootkit
Security Compromise	Yes Suspected No Unknown
System Type	Enterprise Systems Enterprise Systems - Application Layer Enterprise Systems - Database Layer Enterprise Systems - Enterprise Technologies and Support Infrastructure Enterprise Systems - Network Systems Enterprise Systems - Networking Devices Enterprise Systems - Web Layer Enterprise Systems - VoIP Industrial Control Systems Industrial Control Systems - Equipment Under Control Industrial Control Systems - Operations Management Industrial Control Systems - Safety, Protection and Local Control Industrial Control Systems - Supervisory Control Mobile Systems Mobile Systems - Mobile Operating Systems Mobile Systems - Near Field Communications Mobile Systems - Mobile Devices Third-Party Services Third-Party Services - Application Stores Third-Party Services - Cloud Services Third-Party Services - Security Vendors Third-Party Services - Social Media Third-Party Services - Software Update Users Users - Application And Software Users - Workstation Users - Removable Media
Threat Actor Type	Cyber Espionage Operations

	Hacker Hacker - White hat Hacker - Gray hat Hacker - Black hat Hacktivist State Actor / Agency eCrime Actor - Credential Theft Botnet Operator eCrime Actor - Credential Theft Botnet Service eCrime Actor - Malware Developer eCrime Actor - Money Laundering Network eCrime Actor - Organized Crime Actor eCrime Actor - Spam Service eCrime Actor - Traffic Service eCrime Actor - Underground Call Service Insider Threat Disgruntled Customer / User
Sector	Agriculture Arts Construction Consumer Goods Educational Finance Government High Tech Legal Manufacturing Media Medical Non-profit Recreational Service Transportation
Measure Type	Upgrade (software and hardware) Periodically security awareness training for all employees Purchase cybersecurity insurance AAA Develop organizational policy on cyber security Increase cybersecurity employees Collect and save data for investigation Backup and recovery processes Risk assessment Protection Services

Table A.1: Enumerate field content

Appendix B

URLs

This appendix covers all the communication API used in the SHINE system. The parameters written in the form of regular expression were set to match any word character combinations(alphanumeric & underscore). Including these parameters when calling the matching URLs is compulsive and the names for the parameters should be ignored. To be noted, the value for the parameter with name *group* has only two options, namely, *org* stands for *organization* and *sector*. Any other values except *org* and *sector* for the parameter *group* will be responded with a 400 error.

And the parameters that start with a question mark and end with a equal sign are optional, it can either be set with a specific value following the equal sign or be completely ignored, for example, the URL *infoshare/api/AssettypeList/?is_valid=*, it can be used as *infoshare/api/AssettypeList/?is_valid=True* to filter all the types of asset with the field *is_valid* equals to True or 1, or the URL can be used as *infoshare/api/AssettypeList/* to query for all the record for asset type without any filtering processes.

Application and prefix	**URL and paramters**	**Description**
application application/api	add_application/	Add application records to the database
attack_features attackfeatures/api/	AttackFeatureRecordCreate/	Add attack feature records to the database
attack_information attackinfo/api/	MonthlyLossSum/ (?P<group>\w+)/(?P<id>\w+)$	Query for the total loss caused by cyberattacks of a specified organization or sector each month last year

AttackTypeLossSum/ (?P<group>\w+)/(?P<id>\w+)$	Query for the total loss caused by each type of cyberattack on the specified organization or sector in the last year
YearlyLossSumBySector/	Query for the total loss caused by cyberattacks on each sector last year
YearlyLossSumByLossCate/ (?P<group>\w+)/(?P<id>\w+)$	Query for each type of loss caused by cyberattacks in a specific organization or sector last year
MonthlyImpactRating/ (?P<group>\w+)/(?P<id>\w+)$	Query for the count of the number of cyberattacks suffered by a specific organization or sector each month last year according to the impact rating category
MonthlyAttackTimes/ (?P<orgid>\w+)$	Query for the number of attacks suffered by specified organization each month last year
AttackTimesBySector/	Query for the number of cyberattacks suffered by a specified sector last year
overviewIncidentCate/ (?P<group>\w+)/(?P<id>\w+)$	Query for the number of cyberattacks suffered by a specified organization or sector in the last year by each type of cyberattack
MonthlyIncidentCateList/ (?P<group>\w+)/(?P<id>\w+)$	Query for the count of the number of cyberattacks suffered by a specific organization or sector each month last year according to the attack type category

YearlyIncidentEffectTimes/ (?P<group>\w+)/(?P<id>\w+)$	Query for the count of the number of cyberattacks suffered by a specific organization or sector each month last year according to the incident effect category
MonthlySecurityCompromise/ (?P<group>\w+)/(?P<id>\w+)$	Query for the count of the number of cyberattacks suffered by a specific organization or sector in the last year according to the security compromise category
YearlyTopAttackerInfrastructure/ (?P<group>\w+)/(?P<id>\w+)$	Query for the infrastructures used by attackers and the corresponding number of cyberattacks suffered by a specific organization or sector in the last year according to the attack type category and sorted in descending order according to the number of attacks
YearlyTopThreatActorType/ (?P<group>\w+)/(?P<id>\w+)$	Query for the types of threat actors and the corresponding number of cyberattacks suffered by a specific organization or sector in the last year according to the attack type category and sorted in descending order according to the number of attacks
YearlyTopAttackerTool/ (?P<group>\w+)/(?P<id>\w+)$	Query for the tools used by attackers and the corresponding number of cyberattacks suffered by a specific organization or sector in the last year according to the attack type category and sorted in descending order according to the number of attacks

YearlyTopMalwareType/ (?P<group>\w+)/(?P<id>\w+)$	Query for the malware types used by attackers and the corresponding number of cyberattacks suffered by a specific organization or sector in the last year according to the attack type category and sorted in descending order according to the number of attacks
YearlyTopSystemType/ (?P<group>\w+)/(?P<id>\w+)$	Query for the system types affected by cyberattacks and the corresponding number of cyberattacks suffered by a specific organization or sector in the last year according to the attack type category and sorted in descending order according to the number of attacks
YearlyTopLossProperty/ (?P<group>\w+)/(?P<id>\w+)$	Query for the property loss caused by cyberattacks and the corresponding number of cyberattacks suffered by a specific organization or sector in the last year according to the attack type category and sorted in descending order according to the number of attacks
YearlyLossDuration/ (?P<group>\w+)/(?P<id>\w+)$	Query for the time loss caused by cyberattacks and the corresponding number of cyberattacks suffered by a specific organization or sector in the last year according to the attack type category and sorted in descending order according to the number of attacks

YearlyTopIpWatchlist/ (?P<group>\w+)/(?P<id>\w+)$	Query for the IP addresses involved in cyberattacks and the corresponding number of cyberattacks suffered by a specific organization or sector in the last year according to the attack type category and sorted in descending order according to the number of attacks
YearlyTopMaliciousEmailWatchlist/ (?P<group>\w+)/(?P<id>\w+)$	Query for the malicious e-mail addresses involved in cyberattacks and the corresponding number of cyberattacks suffered by a specific organization or sector in the last year according to the attack type category and sorted in descending order according to the number of attacks
YearlyTopFileHashWatchlist/ (?P<group>\w+)/(?P<id>\w+)$	Query for the file hashes involved in cyberattacks and the corresponding number of cyberattacks suffered by a specific organization or sector in the last year according to the attack type category and sorted in descending order according to the number of attacks
YearlyTopDomainWatchlist/ (?P<group>\w+)/(?P<id>\w+)$	Query for the domain names involved in cyberattacks and the corresponding number of cyberattacks suffered by a specific organization or sector in the last year according to the attack type category and sorted in descending order according to the number of attacks

YearlyTopURLWatchlist/ (?P<group>\w+)/(?P<id>\w+)$	Query for the URLs involved in cyberattacks and the corresponding number of cyberattacks suffered by a specific organization or sector in the last year according to the attack type category and sorted in descending order according to the number of attacks
YearlyTopAssetType/ (?P<group>\w+)/(?P<id>\w+)$	Query for the asset types affected by cyberattacks and the corresponding number of cyberattacks suffered by a specific organization or sector in the last year according to the attack type category and sorted in descending order according to the number of attacks
YearlyDiscoveryMethod/ (?P<group>\w+)/(?P<id>\w+)$	Query for the discovery methods of cyberattacks and the corresponding number of cyberattacks suffered by a specific organization or sector in the last year according to the attack type category and sorted in descending order according to the number of attacks
MonthlyLossDuration/ (?P<group>\w+)/(?P<id>\w+)$	Query for the time loss caused by cyberattacks and the corresponding number of cyberattacks suffered by a specific organization or sector each month last year according to the attack type category and sorted in descending order according to the number of attacks
AttackInformationDetail/ (?P<pk>\w+)$	Retrieve one record of attack information according to the primary key

	AttackInformationRecordCreate/	Create one record of attack information
city_country city_country/api/	country_list/	Query for the list of all countries and the details
	city_detail/(?P<pk>\w+)$	Retrieve the details of a city according to the primary key
	city_list/?countryid=	Query for the list of all cities or the cities that belongs to one country
information_sharing info_share/api/	AssettypeList/?is_valid=	Query for the list of all asset types or the asset types that with a specified value for *is_valid* field
	AttackerinfrastructureList/?is_valid=	Query for the list of all types of attacker infrastructure or the types of attacker infrastructure that with a specified value for *is_valid* field
	AttackertoolList/?is_valid=	Query for the list of all types of attacker tool or the attacker tools that with a specified value for *is_valid* field
	ImpactratingList/	Query for the list of all impact ratings
	IncidentcategoryList/	Query for the list of all incident categories
	IncidenteffectList/?is_valid=	Query for the list of all types of incident effect or the types of incident effect that with a specified value for *is_valid* field
	LosspropertyList/?is_valid=	Query for the list of all types of loss property or the types of loss property that with a specified value for *is_valid* field

	MalwaretypeList/?is_valid=	Query for the list of all malware types or the malware types that with a specified value for *is_valid* field
	SecuritycompromiseList/	Query for the list of all types of security compromise
	SystemtypeList/?is_valid=	Query for the list of all system types or the system types that with a specified value for *is_valid* field
	ThreatactortypeList/?is_valid=	Query for the list of all threat actor types or the threat actor types that with a specified value for *is_valid* field
	DiscoverymethodList/?is_valid=	Query for the list of all types of discovery method or the types of discovery method that with a specified value for *is_valid* field
	LossdurationList/	Query for the list of all types of loss duration
measure measures/api/	MeasureRecordCreate/	Create one record of measure
	MeasureRecordList/ (?P<userid>\w+)/ (?P<incident_category>\w+)$	Query for the list of measures set by a specified user for a specified type of attack
	MeasureRecordListByUser/ (?P<userid>\w+)$	Query for the list of all measures set by a specified user
measure_type measuretype/api/	MeasuretypeList/?is_valid=	Query for the list of all types of measure or the types of measure that with a specified value for *is_valid* field

organization organization/api/	org_list/?is_valid=&sector=	Query for the list of all organizations
sector sector/api/	show_sector/	Query for the list of all sectors or create a record of sector depends on the http method
	sector_detail/(?P<pk\w+)$	Retrieve, delete or modify a record of a sector according to the primary key and the http method
	sector_list/?is_valid=	Query for the list of all the sectors or the sectors that with a specified value for *is_valid* field
user_info userinfo/api/	UserDetail/(?P<pk>\w+)$	Retrieve or modify a record of a user's information according to the primary key and the http method
	UserList/	Query for the list of all the users information
	UserCreate/	Create a record of user's information

Table B.1: URLs

Appendix C

Installation Guidelines

C.1 Introduction

This project consist of two parts:

- The *Front End* is mainly inherited from DDoSGrid project, which used Vue.js as their developing language, and containts three parts, the *miner*, the *api* and the *frontend*.
  - The *miner* subproject is inherited from DDoSGrid, which is a packet decoder and feature extractor.
  - The *api* is inherited from DDoSGrid, which is a RESTful api based on Express.js.
  - The *frontend* is a Vue.js based application that renders visualizations obtained from the backend.
- The *Back End* is responsible for data processing, data management, and applications communication. Our Back End system is based on Django. There are 10 applications functioning in the back-end.
  - The *application* application is used to process the join in application of interested persons.
  - The *attack_feature* application is used to handle some data extracted from the PCAP files.
  - The *attack_information* application is responsible for process single incidents related data, such as the financial loss caused by the incident and the technical detail about the incident. This application also in charge of providing data for *Incident Statistic* part and *Ecomonic Impact* part of system.
  - The *city_country* application provides the city and country options for users to choose.

- The *information_sharing* application manages the options for technical related multiple choices and provide users with those options to choose from.
- The *measure* application deals with the countermeasures set by users against cyberattacks.
- The *measure_type* application manages the predefined countermeasures for users to select from.
- The *organization* application is used to handle organization related data and provide users with options.
- The *sector* application is used to handle sector related information and provide options as well.
- The *user_info* application is used to process the user related information.

And within each application, there are six python files been used to make sure the system functioning as designed.

- *admin.py* is related to the settings of Django admin site.
- *app.py* is where the application configuration metadata for an application being stored.
- *model.py* is the place where data models are defined matching with the tables in the database.
- *serializers.py* is the place to define serializers for data models which are reponsible for data validation and conversion.
- *urls.py* is where the URLs configurations being set to dispatch requests to appropriate methods for processing.
- *views.py* is where all the *views* belong to, and each *view* contains concrete implementations of methods to handle requests.

The following shows the outline structure of the SHINE system.

```
|—— api
|—— application
|—— attack_features
|—— attack_information
|—— city_country
|—— docker-compose.ysl
|—— Dockerfile
|—— Economic.db
|—— frontend
|—— information_sharing
|—— LICENSE
|—— manage.py
|—— measure
|—— measure_type
```

```
|—— miner
|—— models.py
|—— organization
|—— README.md
|—— sector
|—— SHINE
|—— user_info
```

C.1.1 Required environments

The following applications and programs must be installed on the target machine before the installation.

- Node.js
- npm
- git
- Python, Version >= 3.6
- pip
- Django 3
- libpcap
- SSH Client

C.2 Front End

Clone the project from github:

```
1  git clone git@github.com:luke-feng/MAP.git
```

C.2.1 Miner

Enter the *miner* subproject and install the necessary dependencies. Make sure you are running Node.JS version 10 and that you lave libpcap installed.

```
1  cd miner
2  npm i
```

After that the miner package can be imported as an NPM module or it can be run manually through the shell. For details of this part, please refer to the DDoSGrid project [10].

C.2.2 API

Before install and start the API, please make sure that you have already installed the dependencies of the miner. And then change into the API directory to install the required modules.

```
1 cd miner; npm i; cd ..;
2 cd api; npm i
```

Now simply run it and optionally pass the port where it should listen:

```
1 node index.js
```

And you can set a listening port as your wish by using:

```
1 export PORT=1234; node index.js
```

Otherwise by using the scripts (recommended):

```
1 ./scripts/start_dev_server.sh
```

You may need to change *CLIENT_APP_ORIGIN* , *OAUTH2_SUCCESS_FORWARD*, *OAUTH2_CALLBACK* to your own IP and service port in the *api/scripts/start_dev_server.sh* script file.

It will automatically listen on *8080* port. And you can change the technical details in the file *api/scripts/start_dev_server.sh*, like the listening port and origin client application.

C.2.3 Frontend

Enter the *frontend* subproject and run it after fetching its dependencies

```
1 npm i
```

This will automatically build the project. To connect it with our *Back End*, you can run the build command:

```
1 npm run build
```

It will packet the frontend project and connect with Django backend.

C.3 Back End

C.3.1 Installing required third-party packages

With the *pip* being installed, you can use this command to install packages. Run the following commands to install required third-party packages for the SHINE system.

```
1 pip install djangorestframework
2 pip install django-filter
3 pip install django-smart-selects
4 pip install django-cors-headers
5 pip install django-unixtimestampfield
6 pip install geonamescache
```

If you would like to install other packages, please do not forget to add them to the *INSTALLED_APP* list inside the *settings.py* file.

C.3.2 Configure the path

For this step, you need have your server IP address and the port number that provides service for the DDoSGrid system in hand. Fine the hidden file *.env.production*, open the file and locate yourself to the line that sets *VUE_APP_APIBASEURL* and *VUE_APP_SHINEBASEURL*, and then modify it follow the following format.

VUE_APP_APIBASEURL = http://*your server IP address*:*DDoSGrid service port*
VUE_APP_SHINEBASEURL = http://*your server IP address*:*BackEnd service port*

C.3.3 Add IP address to the white list

Before you can start run the service, you need to add the server IP address to the trusted white list. To do so, you need go find the file *settings.py* under the SHINE folder. Open the file and find the place where the *ALLOWED_HOSTS* and *CORS_ORIGIN_WHITELIST*, and append your server IP address to the end of the list.

C.3.4 Start the service

To start the server, run the following command within the folder that stores the *manage.py* file.

```
1 python manage.py runsever
```

And the terminal will tell you the which port is currently carrying the service.

Appendix D

Contents of the CD

The CD contains the project source code, database and the installation instructions. Some artificially generated data, which is used to evaluate the platform, is also included.