SEED

(Security Event Enumeration & Detection)

By
NC Syed Hamza Ali Shah
NC Syed Muhammad Hassaan
NC Ali Ahmed Dar
NC Sharjeel Ahmed

Supervisor
Dr. Faisal Amjad

Submitted to the faculty of Department of Computer Software Engineering,
Military College of Signals, National University of Sciences and Technology,
in partial fulfillment for the requirements of B.E Degree in Software Engineering
May 2022

CERTIFICATE OF CORRECTIONS & APPROVAL

Certified that work contained in this thesis titled "Security Event Enumeration Detection (SEED)" carried out by Nc Syed Hamza Ali Shah, Ali Ahmed Dar, Sharjeel Ahmed and Syed Muhammad Hassaan under the supervision of Dr. Faisal Amjad for partial fulfillment of Degree of Bachelor of Software Engineering, in Military College of Signals, National University of Sciences and Technology, Islamabad during the academic year 2018-2022 is correct and approved. The material that has been used from other sources it has been properly acknowledged / referred.

Approved by

Supervisor

_____

Date

_____

DECLARATION OF ORIGINALITY

We hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature of Students

_____

Syed Hamza Ali shah
00000244068

_____

Ali Ahmed Dar
00000244376

_____

Syed Muhammad Hassaan
00000245848

_____

Sharjeel Ahmed
00000249781

Signature of Supervisor

Acknowledgements

We are thankful to our Creator Allah who guided us throughout this work at every step and for every new thought which we setup in my mind to improve it.

We would also like to express special thanks to my supervisor A/P Dr Faisal Amjad for his help throughout my thesis. I can safely say that I haven't learned any other engineering subject in such depth than the ones which he has taught.

I would also like to pay special thanks to Dr Bilal Rauf for his tremendous support and cooperation. Each time I got stuck in something, he came up with the solution. Without his help I wouldn’t have been able to complete my thesis. I appreciate his patience and guidance throughout the whole thesis.

Finally, I would like to express my gratitude to all the individuals who have rendered valuable assistance to my study.

Table of Contents

CERTIFICATE OF CORRECTIONS & APPROVAL iii
DECLARATION OF ORIGINALITY iv
Plagiarism Certificate (Turnitin Report) v
Acknowledgements 6
Table of Contents 7
List of Figures 10
Abstract 11
CHAPTER 1: INTRODUCTION 12
1.1 Problem statement 13
1.2 Proposed Solution 13
1.3 Working Principle 13
1.4 Objectives 14
1.4.1 General Objectives: 14
1.4.2 Academic Objectives: 14
1.5 Scope 14
1.6 Deliverables 15
1.7 Relevant Sustainable Development Goals 15
1.8 Structure of Thesis 16
Chapter 2: LITERATURE REVIEW 17
2.1 Industrial background 17
2.2 Existing solutions and their drawbacks 18
3.1 Log processing 19
3.2 Visualization 19
3.3 Generation of Alerts: 20
CHAPTER 4: SOFTWARE REQUIREMENTS SPECIFICATION 22
4.1 Operating environment 22
4.2 Design and implementation constraints 22
4.4 Assumptions and dependencies 22
4.5 External Interface Requirements 23
4.5.1 User interfaces 23
4.5.2 Hardware interfaces 23
4.5.3 Software interfaces 23

4.6 Communication interfaces 24
4.6 System Features 25
4.6.1 Log In 25
Description and Priority 25
Stimulus/Response Sequences 25
Functional Requirements 25
4.6.2 Addition of Node 25
Description and Priority 25
Stimulus/Response Sequences 25
Functional Requirements 26
4.6.3 Log processing 26
Description and Priority 26
Stimulus/Response Sequences 26
Functional Requirements 26
4.6.4 Visualization 27
Description and Priority 27
Stimulus/Response Sequences 27
Functional Requirements 27
4.6.5 Generation of Alerts: 27
Description and Priority 27
Stimulus/Response Sequences 27
Functional Requirements 27
4.7 System Diagram 28
4.8 Non-Functional Requirements 29
4.8.1 Performance requirements: 29
4.8.2 Safety requirements: 29
4.8.3 Security Requirements: 29
4.8.4 Software quality attributes: 29
CHAPTER 5: DESIGN SPECIFICATION 30
5.1 Architectural Design 30
5.2 Decomposition Description 31
5.3 Design Rationale 32
5.4 Data Description 32
5.5 COMPONENT DESIGN 33

CHAPTER 6: INTERFACE DESIGN 38
6.1 Overview of User Interface 38
6.2 Screen Images 38
6.3 Screen Objects and Actions 39
CHAPTER 7: CODE ANALYSIS 40
7.1 Front End 40
7.2 Back End 41
CHAPTER 8: CONCLUSION 42
CHAPTER 9: FUTURE WORK 43
References & citations 44
11. Code Analysis 39
11.1 Front-End 39
11.2 Back-End 40
12. Conclusion 42
13. Future work 43
14. References & citations 44

List of Figures

Figure 1: System Diagram 27
Figure 2: Activity Diagram 29
Figure 3: Decompostion Diagram 30
Figure 4: Use case diagrm 32
Figure 5:Sequence diagram 36
Figure 6: User interface 37
Figure 6: Screen Images 38
Figure 7: Front End 39
Figure 8: Back end 39

Abstract

In this age of internet due to the advancement in technology a lot of data is being used and transferred from computer to computer through internet so it has become very important to monitor such huge amount of data because unmonitored data can cause a lot of threats. There are a lot of malwares viruses which can harm computer in one way or the other. To tackle this problem, we proposed a solution that will monitor the data traffic that is coming to the network and inside the network. This solution will fetch the data logs and organize them in a readable format in real time environment and if it contains any harmful aspects based on the defined rules for threats, it will notify the admin the admin and alert will be generated about the possible threats. The admin will be notified about the threats he can see and check if he finds the traffic harmful, he can discard it from the network if he finds it okay he can let it pass through the network.

CHAPTER 1: INTRODUCTION

The Technology has revolutionized our way of living. So with the advancement in technology, it is a need of time to get our life style and needs updated with the technology requirements. The solution implemented will help the SOC analysts to tackle the security problems in a much better way.

With this increasing trend of such huge data logs, companies are depending more and more on SIEM solutions for log analysis which is very expensive. However, with the introduction of open source, lightweight and rich featured Search Engine Database models. The proposed System uses an open-source generic search engine, Elastic Search, and other components to process a large number of logs and to detect attacks.

SEED is an information collection and events-based Cybersecurity solution which will help the *SOC* to identify suspicious activity and hence, provide for quick *incident response*. It is the advanced technology to improve *security*, *visibility*, *actionability*, and *posture*, while reducing operational burden. SEED will provide a platform where all logs are gathered and analyzed in real-time. Instead of only tracking users, it also tracks other entities such as endpoints, applications, and networks in order to find threats and visualize the attack

1.1 Problem statement

We deal with a lot of data during our working environment in labs and offices and some of data is very sensitive. With advancement in technology the number of cyber-attacks throughout the world have increased drastically and with a lot of data travelling through the network there are possible threats which can be affect the performance and result in data breaches and cyber-attacks. It is very difficult to monitor all this traffic in run time and identify which of the traffic is malignant, so this issue needed to be addressed to protect the networks from threats.

1.2 Proposed Solution

We have proposed the solution which will provide a platform where all logs are gathered and will organize them in a human readable format where they are analyzed to notify events in real-time. Instead of only tracking users, it also tracks other entities such as endpoints, applications and networks in order to find threats and will generate the alerts based on the defined rules for threats.

1.3 Working Principle

The project works on the objective of threat detection and is based on the principles of log monitoring and event detection – with applications in SoC. The project is deployed using agent and the tools are integrated in it for the threats monitoring like network traffic capture and IDS (suricata). The end product consists of different modules and all the modules are closely inter-related with each other. The modules are listed here:

- Log collection agents
- Log processor module
- Event Detector
- Alert Generator
- Visualizer

1.4 Objectives

1.4.1 General Objectives:

Our objective is to build the software with integrated tools prototype powered by following security principles, providing a smart administrative tool to reduce the burden on soc analyst and easy administration and identification of threats on network traffic.

1.4.2 Academic Objectives:

- Development of SEED (Security Enumeration Detection) - which will meet our FYP requirements
- To implement the knowledge, we learned in this degree
- To lay the foundation of research for the upcoming batches
- To develop a project that contributes for the protection of computer systems which will benefit the society as well.

1.5 Scope

The proposed system will act like SIEM, to get the logs from all the defined sources, with the following features:

- Using open source and free search engines for centralized log searching, storage, and analyzing compatibility instead of using commercial solutions.
- Developing signatures for search engines to automate the detection of attacks.
- The system should be extensive, flexible and support parallel, distributed architecture.

The end product is intended to handle the following tasks including log collection from defined sources such as network monitor, endpoint security tools such as HIDS and windows/Linux OS event logs, firewalls etc. The logs will be formatted to a unified index and forwarded to a central storage for aggregation. These logs will be categorized and then visualized to the analyst for monitoring. Automatic alerts will be generated on basis of a pre-defined policy.

1.6 Deliverables

A Web Application with visualization on a user-friendly interface which plays an optimum role in the security of a network infrastructure by threat detection using modern techniques.

1.7 Relevant Sustainable Development Goals

This project falls under the UN defined SDG that is ‘Industry, Innovation and Infrastructure’ – this solution is intended to provide more stable interruption free service to the customers and a secure infrastructure. Our project provides the security for the systems which is very important in this era of increasing cybercrimes.

1.8 Structure of Thesis

- **Chapter 1**

  Contains the introduction which is explained above

- **Chapter 2**

  Contains the literature review and the background and analysis study this thesis is based upon.

- **Chapter 3**

  Contains the interface requirements of the project.

- **Chapter 4**

  Software Requirement specification

- **Chapter 5**

  Software Design Specification

- **Chapter 6**

  Interface Design

- **Chapter 7**

  Code analysis

- **Chapter 8**

  Conclusion

- **Chapter 9**

  Future work

- **References & citations**.

Chapter 2: LITERATURE REVIEW

A new solution is launched by improving and modifying the features and functionalities of previously launched similar solutions. Literature review is a significant step for development of an idea to a new solution. Moreover, for the development of a solution / product, and for its replacement, related to traffic system, a detailed study regarding all similar projects is necessary. Our research is divided into the following points.

- Industrial Background
- Existing solutions and their drawbacks
- Research Papers

2.1 Industrial background

In this cyber era, one of the major issues faced across the world of internet and computers is cyber threats. Cyber threats have led us to a lot of problems because we are completely dependent on technology and if it is compromised estimates of loss can be beyond expectations, as we discussed in the Problem Statement, so there was a need of a security solution to tackle this problem. We developed a product a very small scale which is scalable and ultimately, results in a big marketplace for Security sector.

Initially, Software Industries were not very concerned about the security aspect but then due to the increasing cyber-attacks it effected the revenues and income of the company as well as their repute. Then, these started working resulting in increase in security solutions to prevent attacks and meet the demand and encouragement for customers/clients.

Hence, SEED provides good market growth and impacts economy directly as it is for the protection.

With the advent of technology, especially the internet, activities such as hacking, and identity theft have become common thing. Cybercrime incidents are increasing. A firewall or a free antivirus software alone is not enough. Firewall functions by making traffic flow decision by inspecting the data packet headers. Unfortunately, the entire packet content is not inspected. firewall, unlike IDS, cannot identify or stop a malicious code that is embedded within normal traffic making the computer system vulnerable to lethal attacks. A complex attack can bypass a firewall undetected.

On the other hand, IDS systems scrutinize the entire content of each packet for possible signs of malicious activity. Besides, the technique of content scanning it also confers intense packet data analysis as compared to a firewall. The most sophisticated attacks are also detected by intrusion detection systems through the integration of familiar protocols such as HTTP. As such, these events are detected using computer security log analysis by comparing them against known signatures or network behaviour analysis. These are set of rules that are used by an IDS to detect suspicious activities such as DoS attacks. For this to be effective real-time monitoring of the events is essential. It enables the maintenance of an updated database system of attack signatures which can greatly enhance the probability of detection of an attack.

For effective log management, the deployment of fully functional subsystems of an IDS system is necessary. Typically, an intrusion detection system runs by three primary sub-subsystems, the Detection Engine, the Packet Decoder and the Logging and Alerting System. Detection Engines are algorithmically functioning programs which are part of an IDS which support a collection of related signatures. They are specifically designed to inspect and define a definite set of values or rules that are allowable based on the protocol of a particular network system.

2.2 Existing solutions and their drawbacks

Different solutions are previously being provided for the security purposes of network traffic monitor, but every product has some pros and cons. SIEM solutions which are already existing and being implemented in industries. But the problem it has it is very expensive to implement and takes a lot of resources. It requires big budget allocation which is not possible for small level enterprises the solution we offer is for small scale which can be scalable to large enterprises as well

2.3 SEED and its feature analysis

- Behavioral Monitoring: It has built-in log management, service obtainability monitoring, net flow analysis, and network packet capture.

- Intrusion Detection: A vital part of SEED is to monitor the network and assets for threats with Network IDS, Host IDS, File Integrity Monitoring, Registry Monitoring, and Rootkit Detection capability and other type of logs.

- Visibility: It provides a comprehensive surveillance covering the entire IT infrastructure and fast-tracked threat detection capabilities. SEED can monitor mischievous activity over a long period and reveals advanced threats which helps greatly in security monitoring of business activities analysis and detection of an anomaly

- Big Data Storage: The system can allow enterprises to make use of any storage system maximally

- Rapid Processing: There is instant processing of data before storage and achieving an actual-time evaluation of events in short distance.

CHAPTER 3: SEED REQUIREMENTS

SEED has the log management abilities and the Correlation of the Logs to get all the software activity, records, network traffic, and user events. IT evaluates data for events correlation and effect on compliance and security posture. Some other great functionalities offered by SEED are scans and actual-time surveillance

Major functionalities of SEED are presented here

3.1 Log processing

The logs being generated by any node in the system require some processing to make them able to be stored, displayed, monitored and analyzed.

The logs undergo several processes which mainly include but are not limited to:

- Parsing
- Formatting
- Extraction
- Sequencing etc

3.2 Visualization

All the logs and other useful information being fetched from the whole network will be displayed on the User Interface for the system. This part is also very important because it keeps the analyst aware of the activities and events inside the system.

3.3 Generation of Alerts:

This functionality performs a vital role which consists of two steps

- Policy defining: Policy would be defined for the alert generation
- Mapping of rules: All activities are being mapped with the defined rules in the policies.

Whenever there is an activity that is against the defined policy an alert is generated so that the admin can take steps to avoid any damage that may happen to the system. A pop-up alert like notification is generated and displayed to the admin on the interface and also embedded on the alert's dashboard

3.4 Elasticsearch

Elasticsearch is a distributed highly scalable and flexible ALv2 licensed (Apache open-source) search engine that is multitenant-capable and was released in 2010. Elasticsearch is a project that is fairly new. It is built on top of Lucene which is quite a mature Java-based search and indexing technology that is open-source.

Elasticsearch runs in a Java application server. However, applications do not have to be written in Java to work with Elasticsearch. This is because it can send and receive data over HTTP in JSON to search, to index and to manage an Elasticsearch cluster.
Elasticsearch is best for applications which are built to handle real-time or similar data that has to be processed and analysed fast such as analytics applications. Many internet businesses and organizations have been using Elasticsearch. These include Netflix, GitHub, Stack Overflow, and Facebook. These use Elasticsearch to handle requirements of agile data processing and data storage.

The main focus of Elasticsearch is to provide powerful and fast search. It also aims to explicitly address the issues of availability and scalability. Elasticsearch is also built for Big Data search as well as a performance which relational databases were not designed to support. Elasticsearch provides structured search, aggregations, highlighting of hit word and more. These features enable developers and users to extract information that is valuable from their data without regarding the form.

The main features of Elasticsearch include:

- It is open source distributed, highly available and scalable as well as a real-time document store.
- Elasticsearch provides real-time capabilities for search and analysis
- Elasticsearch provides a RESTful API to do lookup, and various other features
- including geolocation, multilingual search, autocomplete contextual suggestions,
- and snippets of results
- Elasticsearch is horizontally scalable and provides easy integration with cloud

3.5 Kibana

Kibana is an open-source platform for data visualization platform that is Apache 2.0 licensed. It helps to visualize all kinds of structured and unstructured data that has been stored using Elasticsearch indexes. Kibana is written in HTML and JavaScript. Kibana uses the powerful indexing and search features of Elast icsearch which are exposed through its RESTful API to present powerful graphics to users. Kibana exposes data using beautiful graphs, geo maps, pie charts, histograms, tables and more. Kibana simplifies understanding large data volumes. A simple browser interface enables quick creation and sharing of dynamic dashboards which display queries real time.

Some of the key features of Kibana are as follows: -

- Kibana provides a flexible visualization and analytics platform for use in business intelligence.
- Real-time analysis, charting, summaries and debugging
- An intuitive, user-friendly interface. The user interface is highly customizable through drag and drop when needed
- Allows saving a dashboard, as well as managing several dashboards
- Dashboards are easily shared and embedded within other system

3.6 Conclusion

In conclusion, though firewalls, antiviruses and router-based packet filtering are essential entities of a general network security environment, they are inefficient on their own. It is, therefore, necessary to effectively manage these security systems such that they complement each other in the efforts to thwart any malicious attacks. IDS tools are being integrated to inside and outside firewalls and are quickly becoming the gold standard in the maintenance of secure network systems. Commercial SIEM's are made for commercial purposes and not provided specific purposes for the organization. Usually commercially available SIEM and log analysis solutions that provide collection and searching applications target a large audience. So difficult to be managed, mold in specific organizational needs. Specific organizational needs some processes to be molded and managed accordingly. While Opensource solutions in this market are largely frameworks and provides a access methods to be built upon them. Some tools that have been built upon them; their framework cannot be integrated into operations without further development. So the need for log search and collect and a customized without limitations search app like Elasticsearch is needed.

CHAPTER 4: SOFTWARE REQUIREMENTS SPECIFICATION

4.1 Operating environment

- **Hardware:**
  - Database Server
  - Custom Firewall
- **Software:**
  - It can be run on both linux and windows operating system
  - Network Infrastructure

4.2 Design and implementation constraints

- The software is intended for security purposes only.
- The software is implemented in English language only
- The software is designed for only limited types of threats

4.3 User documentation and product design specifications:

The user will be able to use the following as guides for using the software:

- User Manual that contains textual and pictorial help for users in guiding them to use the software correctly and troubleshoot it.
- A webpage in the website interface for the software that answers frequently asked questions and has a guide on using the software
- Project Synopsis

4.4 Assumptions and dependencies.

- SEED solution resides in a controlled access facility that prevents unauthorized access.
- There are one or more competent individuals assigned to manage the system.
- Authorized administrators who manage system are trained to use, configure, and maintain, the software.

4.5 External Interface Requirements

4.5.1 User interfaces

The front-end user interfaces will have the following main screens available to the user:

- Launch Screen: Once the application is started, the first page that is displayed is the launch menu which gives a notification that the application is running.
- Main Menu: This is the main menu of the application; it will show different applications that have been deployed in the system and are ready to use.
- Dashboards: The application will have multi-level and different types of dashboards for visualization depending on the type and warning level of the event/threat. The user can interact with this application by pressing the buttons for required functionality scans.

4.5.2 Hardware interfaces

- Database Server
- Custom Firewall

4.5.3 Software interfaces

- **Python:** We will be using python programming language because its more interactive support to our project.
- **JavaScript:** We will use JavaScript for developing web pages
- **Visual Studio:** We will use Visual Studio as a platform to run our python program and web pages
- **Operating System**: Linux for its user-friendliness and compatibility
- **Development**
    - Back-End
    - Front-End
    - Environment
    - Platform
    - Streaming etc.

4.6 Communication interfaces.

The internal communication of the system should be continuous and secure so that the system cannot be compromised, and it also performs efficiently. The system will user secure channels for connections and streaming.

4.6 System Features

4.6.1 Log In

Description and Priority

At every session, the user will be required to authenticate their identity with the credentials set up at the sign-up page.
This is a basic step to allow user to monitor and analyze the data streams.

Once application starts, the login page is displayed which asks the admin credentials if they are right the application is logged in and is ready to start. The priority of this is high, since it is necessary for the authentication of admin.

Stimulus/Response Sequences

Opens up the main dashboard

Functional Requirements

- Connected to network: The admin of the system will be connected to the network after he is logged in
- Providing Credentials: the admin will have to enter credentials to get registered

4.6.2 Addition of Node

Description and Priority

This is a high priority and important feature that enables the addition of node by installing the agent on the node so that it can connect to the central server for monitoring where logs will be transferred and monitored by the admin.

Stimulus/Response Sequences

The node is then connected into the network and the node-server communication/exchange has been started.

Functional Requirements

Join the network: For a new user(admin), there will be a signup option to access the system.

4.6.3 Log processing

Description and Priority

The logs being generated by any node in the system require some processing to make them able to be stored, displayed, monitored, and analyzed. Priority is medium.

Stimulus/Response Sequences

The logs undergo several processes which mainly include but are not limited to:

- Parsing
- Formatting
- Extraction
- Sequencing etc

Functional Requirements

- **Agent Installation**: Agent will be Installed, up and running
- **Logs transfer:** Logs will be transferred for processing

4.6.4 Visualization

Description and Priority

All the logs and other useful information being fetched from the whole network will be displayed on the User Interface for the system. This part is also very important because it keeps the analyst aware of the activities and events inside the system.

Stimulus/Response Sequences

Logs are displayed on the designed dashboards category-wise.

Functional Requirements

- Fetching of logs: Logs shall be fetched
- Processing: Logs will be processed
- Transfer channel: The transfer channel should be working
- Categorization: Logs will be categorized and displayed accordingly

4.6.5 Generation of Alerts:

Description and Priority

Whenever there is an activity that is against the policy an alert is generated and so this task has a high priority so that the admin can take steps to avoid any damage.

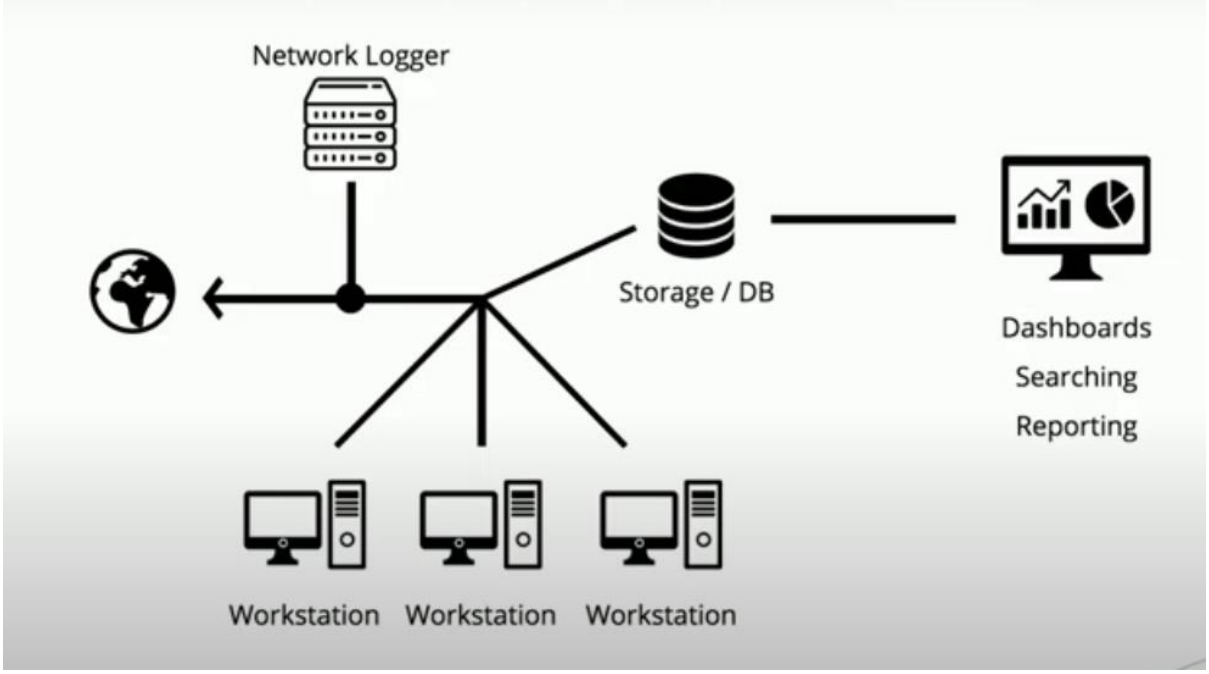
Stimulus/Response Sequences

A pop-up alert like notification is generated and appended to the interface.

Functional Requirements

- Policy defining: Policy would be defined for the alert generation
- Mapping of rules: All activities will be mapped with the defined rules in the policies.

4.7 System Diagram



This is the overview of how our system will look like the right side screen is the admin node where he can see records on his screen which are fetched by database/storage it gather information from the connected computers in the network and the traffic from the internet there is a network logger between it.

4.8 Non-Functional Requirements

4.8.1 Performance requirements:

The system should be capable of handling large amounts of data transferring through its communication channel. There should be a good storage capacity, and there should not be a bottleneck.

4.8.2 Safety requirements:

The use of the software product has no harms on the users; nor does it have any possibility of loss or damage that might be inflicted however during the use of the application,

The data inside the SEED should be made to be safe by implementing adequate security techniques.

4.8.3 Security Requirements:

SEED should ensure that only authorized administrators are granted access to the monitoring of security functions, configurations, and associated data

4.8.4 Software quality attributes:

- **Correctness:**

  The logs generated should always have correct data about threats.

- **Reliability:**

  In the event of a failure, the data on server should stay secured.

- **Adaptability:**

  The server should run on windows OS as well as Linux.

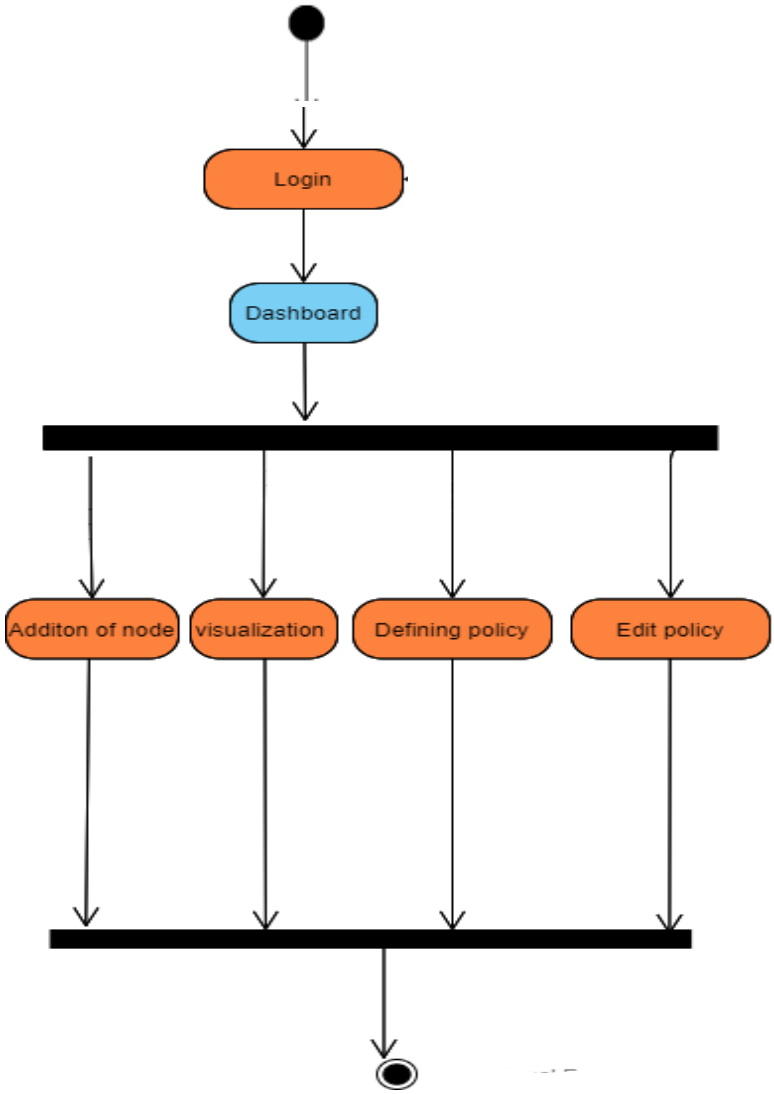
- **Ease of Use**

  The system should be easy to use. The admin will need training of one day to completely understand the system.

CHAPTER 5: DESIGN SPECIFICATION

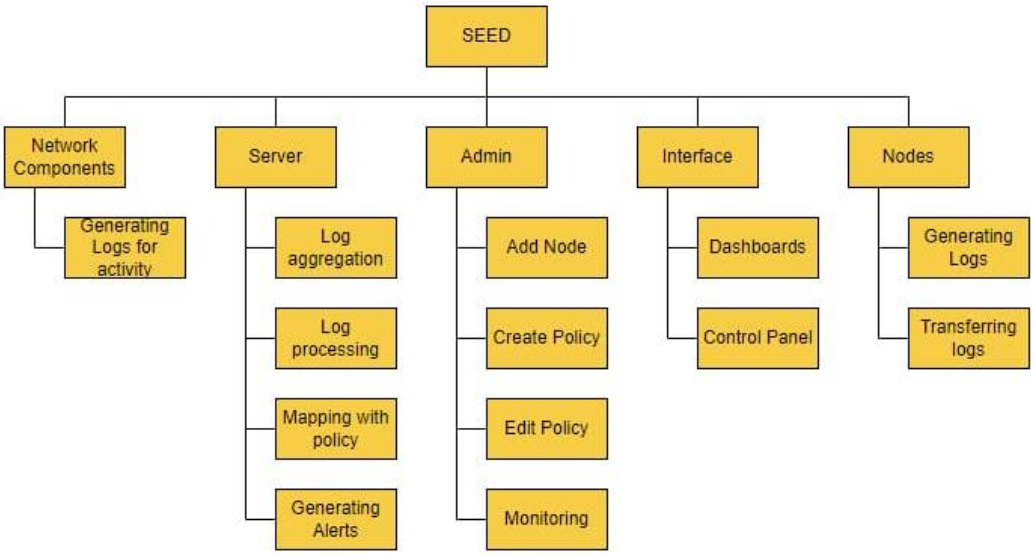
5.1 Architectural Design

The architectural design of SEED is an event-driven architecture. SEED will be continuously monitoring the activities in the system and generate alerts in case of any events occurring.



This is activity diagram of the system which gives the activity details of SEED in which first of all admin logins to the system there is a dashboard screen there are options for him where he can add node to the system, visualize, define policy or edit it.

5.2 Decomposition Description



This is the system decomposition diagram which decomposes the system into sub modules as shown in the figure

The system is overall decomposed into two main components.

5.2.1 User

- **Admin**
  Admin is the user directly interacting with the system through the Human Interface of the system.

- **Nodes**
  Nodes are the end systems and network devices of the system which are added by the admin and send logs to the central server.

5.2.2 Server

Server is responsible for collection, aggregation and processing of the logs to display on the interface (dashboards) of the system and generate alerts after mapping the activities with the policy defined by the admin.

5.3 Design Rationale

The architecture chosen for SEED is Model-View-Controller architecture. MVC architecture is chosen because it provides an effective way of implementing an android application. MVC architecture increases the code testability and makes it easier to implement new features as it highly supports the separation of concerns. Separation of concern means separating an application into distinct sections, User Interaction System, Database, and log processing, so each section addresses a separate concern.
Furthermore, adding a new type of views is very easy in MVC as Model (Database) does not depend on the view. So, any changes in the Model will never affect the entire architecture.

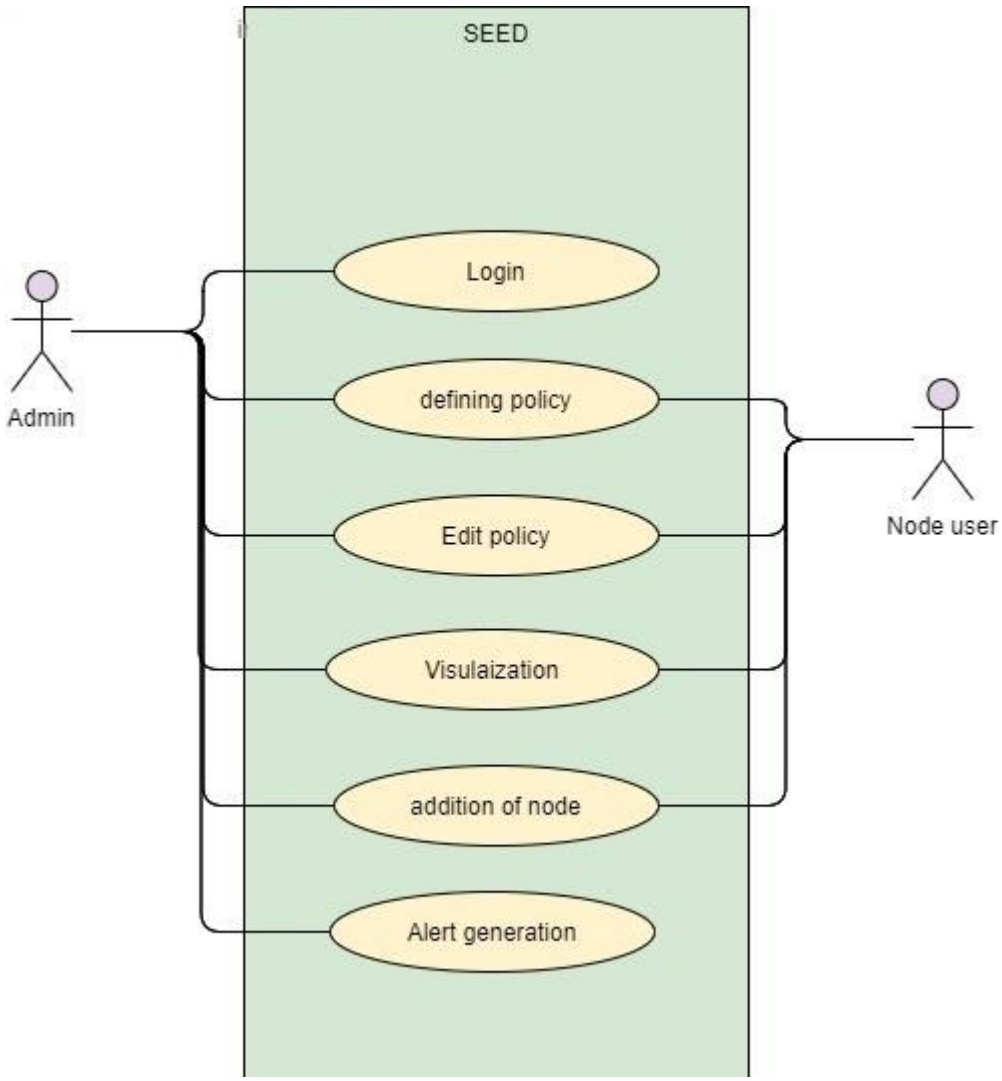
5.4 Data Description

Our system uses a single central database to collect and store all the data coming from all the nodes and devices in the network. This database is located at the central server.

5.5 COMPONENT DESIGN

In this section, we take a closer look at what each component does in a more systematic way.

5.5.1 USE CASE DIAGRAM:



This is the use case diagram of the in which admin and the node user have interactions. The admin can login by entering his credential define policy edit it and can visualize the traffic. He can add another computer to the network which is addition of node by clicking on It and can generates alert.

USE CASE DESCRIPTION:

Use case id	**1**
Use case name	Login
Primary actor	Admin
Secondary actor	N/A
Normal course	User navigates to the login page. User enters the credentials. User clicks on 'login' Option The user is logged in.
Pre-Condition	User's account exists. The server is running.
Post-Condition	The user is navigated to the dashboard.
Alternate course	N/A

Use case id	**2**
Use case name	Addition of node
Primary actor	Admin
Secondary actor	User node
Normal course	User navigates to the Dashboard on the web page. User selects 'Add node to the network' Option User is directed to the node page where he enters the detail. The target node is added.
Pre-Condition	User must ensure that the user node is inside the network and available for the connection
Post-Condition	Node is now connected to the system and starts communication with the server

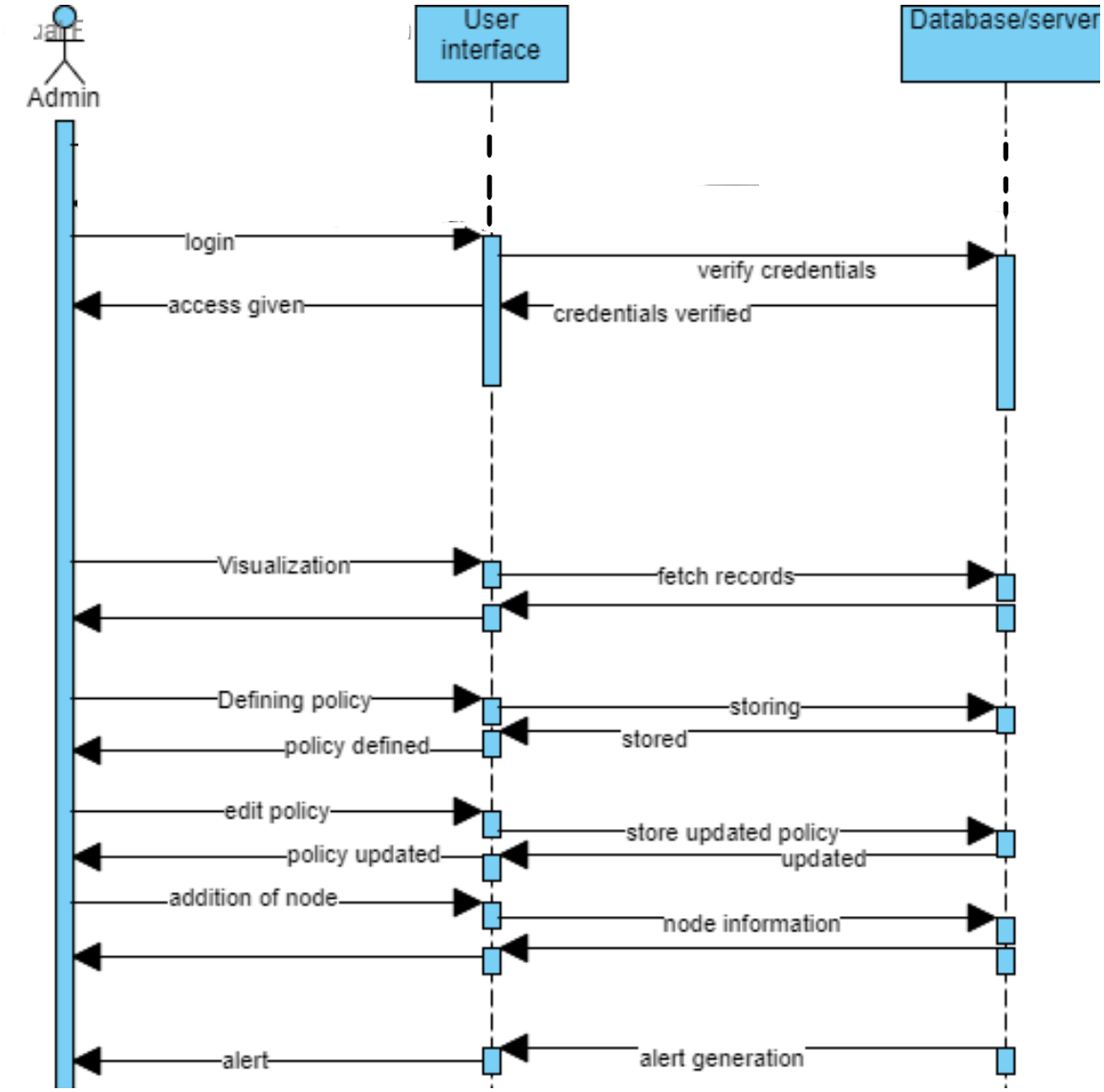
Alternate course	N/A

Use case id	**3**
Use case name	Log processing
Primary actor	User
Secondary actor	User node
Normal course	User navigates to the Detection page on the web page.
Pre-Condition	The nodes are connected and transferring logs to the central server
Post-Condition	Logs have been collected, sanitized for specific features and processed to display on the user interface.
Alternate course	N/A

Use case id	**4**
Use case name	Visualization
Primary actor	Admin User
Secondary actor	User node
Normal course	User navigates to the Dashborad page. User selects ‘Visualization’ Option A screen is displayed where user sees the traffic visualization on the network.
Pre-Condition	The data is being sent to the user interface from the server.
Post-Condition	Information from the server is being displayed on the interface for visualization
Alternate course	

Use case id	**5**
Use case name	Generation of alerts
Primary actor	Admin User
Secondary Actor	User node
Normal course	Logs sent from the nodes are mapped at the server with defined policy for suspicious events. If an event is marked as suspicious, an alert is generated.
Pre-Condition	Logs are being transferred to the server
Post-Condition	Alert is generated and displayed.
Alternate course	

5.6 PROCESS SEQUENCE



This is the sequence diagram of the system in which admin logs in to the system if credentials are right access is given to him then on the dashboard there are multiple options available where he can visualize the network traffic, define policy, edit it add node to the network and generate alerts.

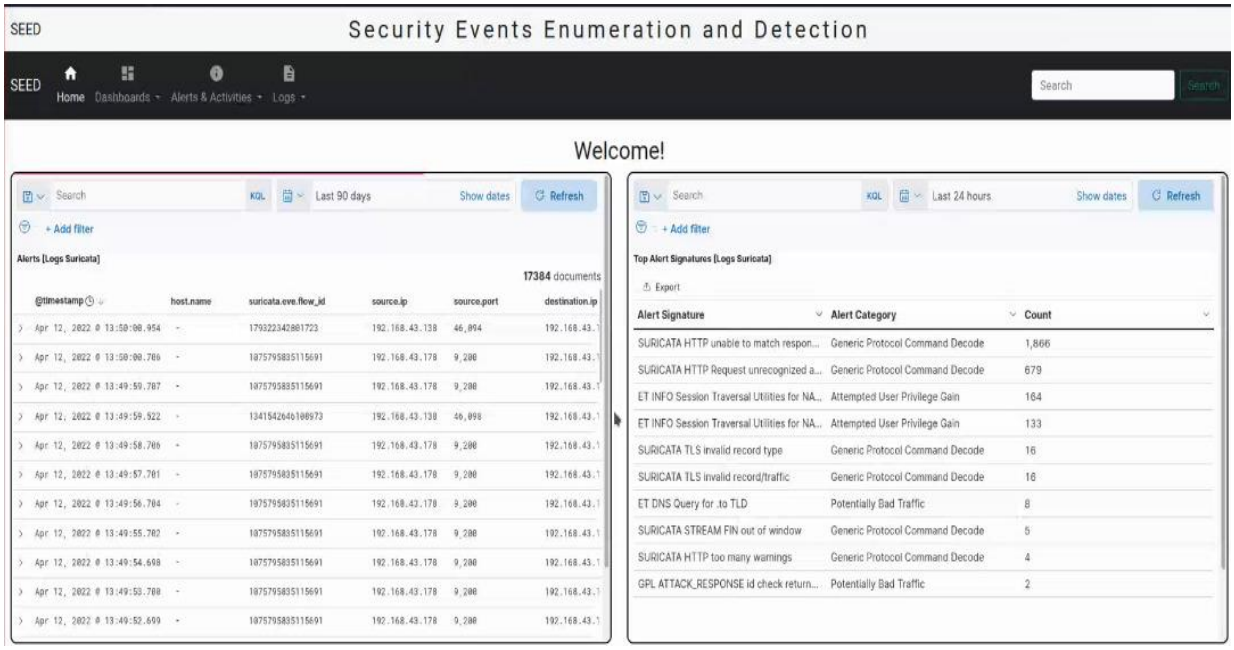
CHAPTER 6: INTERFACE DESIGN

6.1 Overview of User Interface

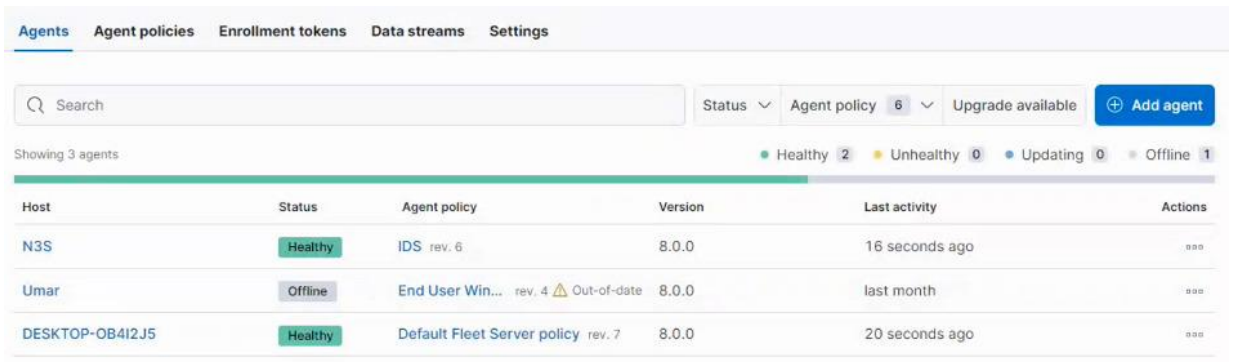
There is only 1 user interacting directly with the system – the admin user. Admin can use the defined functionalities after getting access to the system.

6.2 Screen Images

The front end of our system is presented in the following pictures and this interface is running based on a website.



This figure shows the activity generated by Suricata IDS installed on a host. The dashboard on the left is a record of all the activity going on in the network mentioning the ips, ports and event ids based on timestamp. While the one on right gives a brief but very useful overview of the alert generated by Suricata IDS along with the category and count.



This is the admin interface where the admin can create custom (policy based) agents to be deployed on the hosts in the network. It shows the status of the devices in the network.

This is another example of a dashboard displaying event logs from a host mentioning the ips, ports and protocols along with the timestamp which can be very helpful in monitoring the activity in the network.

This dashboard visualizes the metrices and health status of a node in the network.

6.3 Screen Objects and Actions

The interface has several objects/actions displayed as buttons, tabs or dashboards on the screen. Some of them are just command-oriented while some are switch buttons between different types of dashboards or screens on the web application

CHAPTER 7: CODE ANALYSIS

7.1 Front End

Following is the basis of our front-end code base which runs our web-based GUI.

```
<div>
    <section class="suricata-main">
        <h2>Suricata Alerts</h2>
        <div class="container-fluid">
            <div class="row">
                <div class="col-lg-6">
                    <h3>Signatures</h3>
                    <div class="div-frame"><iframe class="iframe" src="http://192.168.43.178:5601/goto/59bac4c0-9a56-11ec-af37-815d0503e816"></iframe></div>
                </div>
                <div class="col-lg-6">
                    <h3>Visuals</h3>
                    <div class="div-frame"><iframe class="iframe" src="http://192.168.43.178:5601/goto/6be673a0-9a57-11ec-af37-815d0503e816"></iframe></div>
                </div>
            </div>

        </div>
    </section>
</div>
```

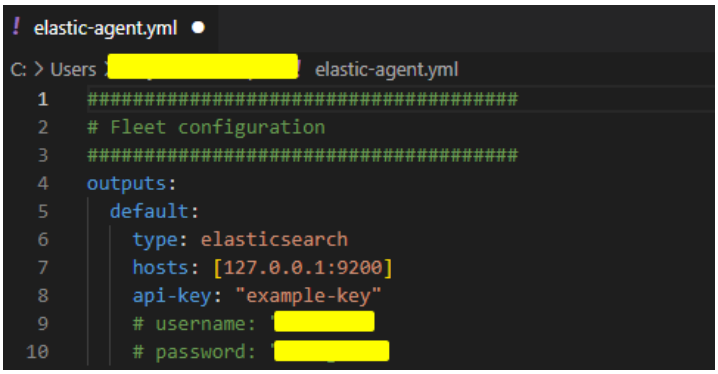
This figure shows the code for the part of webpage where the Signatures of Suricata IDS are displayed.

```
<div class="collapse navbar-collapse" id="navbarSupportedContent">
  <ul class="navbar-nav me-auto mb-2 mb-lg-0">
    <li class="nav-item">
      <a class="nav-link active" aria-current="page" href="#">
        <span class="material-icons-round">home</span>
        <br class="hidden-xs">
        Home
      </a>
    </li>

    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">
        <span class="material-icons-round">dashboard</span>
        <br class="hidden-xs">
        Dashboards
      </a>
      <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
        <li><a class="dropdown-item" href="./suricata.html">Suricata</a></li>
        <li><a class="dropdown-item" href="#">Dashboard 2</a></li>
        <li><hr class="dropdown-divider"></li>
        <li><a class="dropdown-item" href="#">Something else here</a></li>
      </ul>
    </li>

    <li class="nav-item dropdown">
      <a
        class="nav-link dropdown-toggle"
        href="#"
        id="navbarDropdownMenuLink"
        role="button"
        data-bs-toggle="dropdown"
        aria-expanded="false"
      >
      <span class="material-icons-round">info</span>
      <br class="hidden-xs">
        Alerts & Activities
      </a>
      <ul class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
        <li>
          <a class="dropdown-item" href="#">
            Alerts &raquo;
          </a>
          <ul class="dropdown-menu dropdown-submenu">
            <li>
              <a class="dropdown-item" href="./suricata_alerts.html">Suricata Alerts</a>
            </li>
            <li>
```

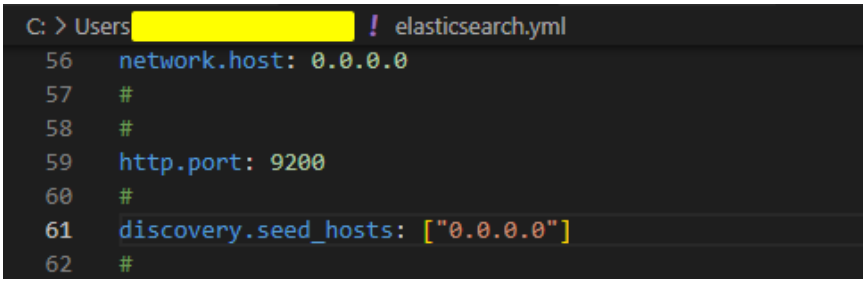
This snippet of the code is a portion of the main home screen of our website.

7.2 Back End

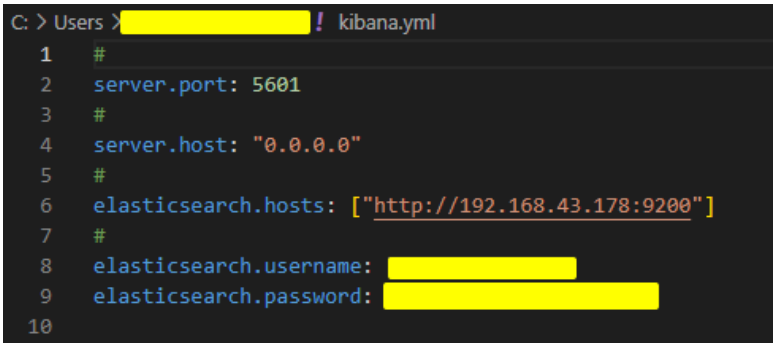
The following code snippets show the connectivity of the prototype network infrastructure.



This picture shows the configuration of the agent installed in the node which is used for forwarding of logs to the central server.



This is the default configuration of elasticsearch server module running on localhost port number 9200.



This is the Kibana module configuration mentioning that Kibana runs on localhost port 5601 and is accessible from within the network conditioned with credentials. The Kibana visualizer gets its input data from the Elasticsearch server.

CHAPTER 8: CONCLUSION

We have developed a prototype with basic functionality that covers the basic features and functionalities required for a security solution to protect the network. We have completed the requirements of the project. The project is expandable and has the capacity to meet the needs of any type of IT infrastructure including computer laboratories, IT businesses and corporate sector.

CHAPTER 9: FUTURE WORK

There is a lot of work that can be done and will be done on this technology stack and the project. We have touched the basics and tried our best to implement it in the given time frame. This project has provided the basics and ideas for the future implementors to work on Elasticsearch and add functionalities and modules to the system and improve the existing system. A major improvement can be the implementation of a suitable machine learning model to automatically take actions based on alerts generated by this product. The machine learning implementation will also be vital in quick detection of any malicious activity on the system.

References & citations

- IBM Security. IBM QRadar SIEM. White Paper. Available online: https://www.ibm.com/downloads/cas/RLXJNX2G (accessed on 12 December 2020).
- Quest. SIEM Integration Best Practices: Making the Most of Your Security Event Logs. White Paper. Available online: https://www.quest.com/whitepaper/siem-integration-best-practices8139415/ (accessed on 31 May 2021).
- CA Enterprise Log Manager. Administration Guide. Available online: https://ftpdocs.broadcom.com/cadocs/0/CA%20 Enterprise%20Log%20Manager%20r12%201%20SP3-ENU/Bookshelf_Files/PDF/CAELM_Admin_ENU.pdf (accessed on 31 May 2021)
- McAfee. Security Information and Event Management (SIEM), Official Website. Available online: https://www.mcafee.com/ enterprise/en-us/products/siem-products.html (accessed on 12 February 2021).
- Trustwave. SIEM Enterprise, Product Brief. Available online: https://trustwave.azureedge.net/media/13581/tw-siementerprise.pdf?rnd=131659475410000000 (accessed on 12 February 2011).
- TIS-233_R-ELK stack for log analysis using customized ids signature
- https://infosecwriteups.com/building-a-siem-combining-elk-wazuh-hids-and-elastalert-for-optimal-performance-f1706c2b73c6

□ j

ORIGINALITY REPORT

6%	5%	0%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	docplayer.net Internet Source	2%
2	www.securonix.com Internet Source	1%
3	Submitted to University of Wales Institute, Cardiff Student Paper	1%
4	Submitted to University of Greenwich Student Paper	1%
5	www.geeksforgeeks.org Internet Source	<1%
6	Submitted to Atilim University Student Paper	<1%
7	Eliza Gomes, Daniel Penz, Viviane Etges Gomes, Carlos Roberto De Rolt, Mario Dantas. "Evaluating the tools to analyze the data from the ParticipACT Brazil Project: A test with Elasticseach Tools Ecosystem with Twitter data", 2018 IEEE Symposium on Computers and Communications (ISCC), 2018 Publication	<1%
8	haritibcoblog.com Internet Source	<1%
9	www.papercamp.com Internet Source	<1%
10	www.coursehero.com Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		