Cyber Incident Response Plan Objectives and Evaluation

System Architecture Design and Research Report

Project Team - 02

Name	Position	Email	Phone
Aidhan Mitsopoulos	Scrum Master	103598809@student.swin.edu.au	0428770628
Habib Mustafawi	Product Owner	102053200@student.swin.edu.au	0466368916
Numil Fernando	Development Team Member	103517163@student.swin.edu.au	0406164700
Thomas Davis	Development Team Member	103203475@student.swin.edu.au	0449619570
Huy Tran	Development Team Member	102559614@student.swin.edu.au	0403204649
Zahin Un Nafi	Development Team Member	103539510@student.swin.edu.au	0481834630

SWE40001, Software Engineering Project A, Semester 1 2023

Document Change Control

Version	Date	Authors	Summary of Changes
1.10	02/05/23	Whole team	Creation
1.20	03/05/23	Whole team	Section 1 done
1.30	04/05/23	Whole team	Section 2 done
1.40	05/05/23	Whole team	Section 3 done
1.50	06/05/23	Whole team	Section 4 done

Document Sign Off

Name	Position	Signature	Date
Aidhan Mitsopoulos	Scrum Master	Aidhan	06/05/23
Habib Mustafawi	Product Owner	Habib	06/05/23
Numil Fernando	Development Team Member	Numil	06/05/23
Thomas Davis	Development Team Member	Thomas	06/05/23
Huy Tran	Development Team Member	Huy	06/05/23
Zahin Un Nafi	Development Team Member	Zahin	06/05/23

Client Sign off

Name	Position	Signature	Date
Dr. Rory Coulter			
Organisation			
Retrospect Labs			

1. Introduction

Within this document, the architecture of the Gradio AI model that will be built will be showcased to any reader needing an understanding of the framework for our Open AI model operating with Gradio's GUI. The purpose of this model is to thoroughly explain and highlight the aspects of our final product's architecture and how each of them interconnect.

1.1 Overview

This report is to summarise the general architecture of Group 2's Gradio CIRPAE model. Highlighting the models design, how each aspect of the model interconnects between themselves. This report will go over this systems Architecture from modules 3-4.

1.2 Definitions, Acronyms and Abbreviations

CIRPAE = Cyber Incident Response Plan and Evaluation
Gradio = The OpenAI model we are training and manipulating
SRS = System Requirements Specification Document

2. Problem Analysis

2.1. System Goals and Objectives

The high-level system goal expected for the Gradio AI is an artificial intelligence that can be parsed large amounts of CIRP playbooks and can grow in its understanding of how to handle any form of Cyber breach. With the user being able to ask the AI through an input box that will then be capable of using it's understanding from its vast knowledge library and providing a detailed output that describes a well-structured response, see figure 3 and 4 in the SRS for more detailed explanations of the functional and non-function goals expected of the product.

2.2. Assumptions

In our assumption, we expect the reader to be well versed in the many different models for explaining System Architecture, these being the Component and Connector view and the Deployment and allocation view, with many others. We do this to avoid over explaining the model and why we either chose to use and or abandon such architectural designs.

Another assumption made, is that the reader is also capable of accessing our numerous other documents when referenced, for their sake we do not explain aspects that our other documents go into far more detail, as such we reference, we're to find better explanations and diagrams to support the context for this Document.

3. High–Level System Architecture and Alternatives

The software system will be developed following a mix of client-server and rule-based architecture.

1. Client: A web-based user interface that allows users to interact with the system.
2. Server: A backend system responsible for processing user requests and generating responses. This will be done using Langchain, which is a framework built around LLMs (Large Language Model).
3. Database: A database that will hold CIRP's and our custom knowledge base (KB).

Rule-Based Architecture:

A Rule-Based system will use conditions to determine an action, these are called rules that are human made to store, sort and manipulate data. For example, if a CIRP had trigger words (condition) that would be considered bad/not useful, then it would mark (action) that CIRP as not helpful. A custom knowledge base will store the rules (conditions) which will determine the actions.

3.1. System Architecture

Component and connector view

- User Interface component: This component provides the user interface for the AI model which is run through Gradio. It communicates with other components to retrieve and display data when certain information has been inputted by a user.
- Data storage component: This component will store all the CIRP custom knowledge base which has been inputted. These documents will be used by the CIRP model to display relevant answers to their requested question. This communicates with the User interface component to provide data as it is being requested.

The connectors between components

- User interface to Data storage container: This connector will allow the user interface to communicate with data storage to access, read and retrieve data relevant to the problem.

Deployment allocation view

Input Box to Gradio Local host: This connector allows the User to ask the AI any form of question relating to its intelligence of CIRP's.

- Gradio Local Host to Knowledge Library: With this upon receiving a question from the User, the AI will cipher through its knowledge library and create a detailed response that will be pushed into the Output Box.
- CIRP Parsing to Knowledge Library: this provides the user with the capabilities of updating the knowledge Library of their model, to further increase it's intelligence and thereby improving the outputs of the AI.

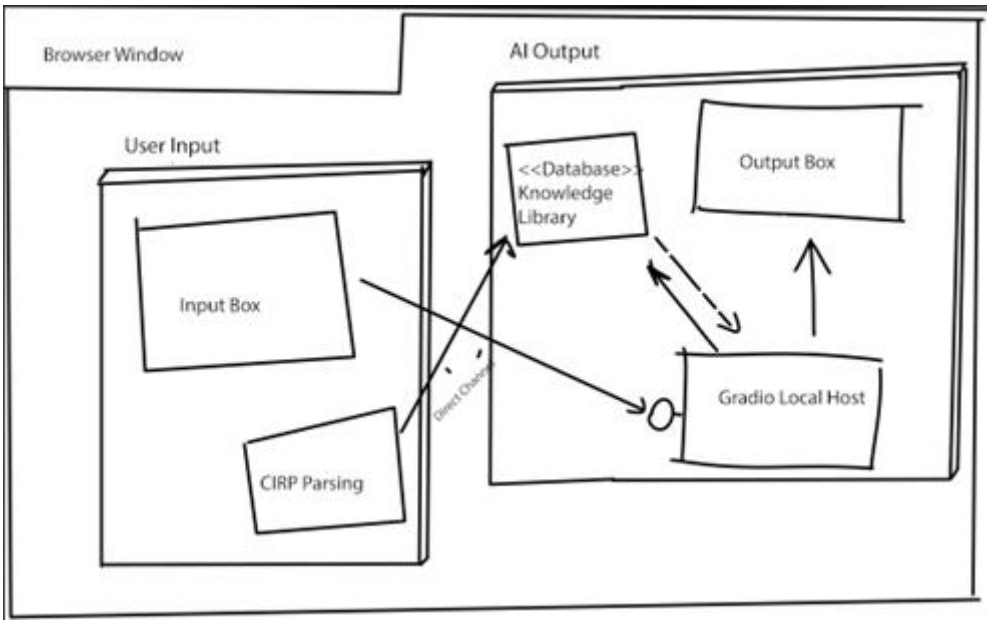

Figure 1: Deployment allocation view

Justification:

The use of the Deployment Allocation View was chosen due to its simple yet descriptive layout of a deployed software, that even inexperienced users can read and understand. In this diagram, one can see the local hosts deployment and how each aspect of the model interacts and affects each other, thereby exploring what aspect interacts with each other, resulting in the final output.

3.2. Other Alternative Architectures Explored

Object: While useful in certain aspects, it is completely inferior when in comparison to the Deployment Allocation view when used for this project. Due to the sheer size of the AI, being able to map out all of its objects and allocating their connections as such would prove fruitless as the complexity of the diagram would be too much for even an experienced reader to comprehend in a short amount of time. Moreover, as we are training the model and implementing our own Knowledge Library, the Objects of the model are not being altered but rather what they shall be interacting with.

Component: This may be the most similar to our chosen architectural diagram, however one aspect made this be seen as the inferior model. This flaw is it requiring to showcase the complete structure of the model, including all of its classes and interfaces of the product. Due to the very nature of the model, like the Object class, this could not be chosen, even if it described the inner structure. Therefore, even if it could describe everything in detail, the diagram would be a jumbled mess and would not describe any aspect of the product to a worthy extent.

4. Research and Investigations

The research that was undertaken to design the plan of the software system could be split into three stages. First, it was to understand the background of the project. We had to understand the context that was given to us by the client - what are the values and motivations of our client's company and how our project would support these values. Retrospect Labs aims to deliver improve their client's cyber security readiness by providing them with common cyber incident exercises. From there we would research on what a Cyber Incident Response Plan (CIRP) was and understand how these plans would help our client. Second, we would need to research into what AI model would allow us to develop a system that would be able to accept information and then be able to respond to any user inputs, giving us the desired output. This would allow for the software system to analyse any inputs that is given to it and determine whether they have met the standards of the CIRP's it's been fed. Finally, we would need to be able to present this to the user via a graphical user input, making sure that it is easy to use and read.

4.1. Research into Application Domain

The cybersecurity sector serves as the project's application domain. This sector is concerned with guarding computer systems and networks from intrusions, assaults, and data breaches. The project team researched cybersecurity and existing cybersecurity incident response strategies to better understand this area and its unique requirements.

Studying the numerous cyber risks and assaults that organisations encounter as well as the typical defensive tactics and strategies was part of the research into the cybersecurity field. For a complete grasp of the cybersecurity landscape, this entailed examining industry publications, research articles, and other pertinent sources.

The project seeks to use technology for natural language processing (NLP) to extract goals from incident response plans and assess their efficacy. In order to accomplish this, the team gathered publicly accessible incident response plans and utilised them to train a machine learning model that can extract important data and insights from the plans. In order to include them into our knowledge base, we did research on existing cybersecurity playbooks and scenario manuals. To create datasets, we are gathering PDFs from various cybersecurity information sources. We have gathered data from sources like CISA, AWS, and other incident response playbooks that are openly accessible online.

4.2. Research into System Design

To choose the best architecture and programming languages for the creation of the Cyber Incident Response Bot, research into system design was undertaken. Python was chosen as the main programming language for this project because of its adaptability and simplicity. Python was a great choice for our project because it is a widely used language in the machine learning and NLP areas.

Natural language processing (NLP) is used by the Cyber Incident Response Bot in this project using the OpenAI NLP engine. We use the Open AI codebase using an API key given by the OpenAI organisation in return for tokens purchased; this is a paid service as the Open AI codebase is not publicly accessible. OpenAI's GPT-3 architecture was our choice for this project, given that it is a particularly potent language model with the capacity to produce human-like replies while being more cost-effective than the other OpenAI modules.

A unique knowledge base for the chatbot was developed using Langchain, a Python module for natural language creation. In order to build this knowledge base, text files and PDFs pertaining to cybersecurity playbooks and scenario training exercises were imported, and Langchain was then trained to provide replies based on this knowledge.

Another Python library called Gradio was used to create the chatbot's user-friendly interface. Gradio makes it easy to host the application on the web and offers users a straightforward and intuitive web-based interface to communicate with the chatbot.

These components were combined to create a strong and effective system that conversely offers dynamic responses to cybersecurity problems and circumstances.

4.3. Research into technical platforms, languages and tools

The project being created will incorporate a range of tools / resources that all play a vital role in the working program. The programming language chosen is Python, but other tools and resources being called upon are Gradio and OpenAI, specifically the Davinci base model.

Gradio is a Python library that allows us to create custom user interfaces for machine learning models. These interfaces allow users to interact with our model in real time being able feed text in and receive an output instantly.

OpenAI's model GPT-3 provides us with a range of models that can understand and generate natural language. We have chosen to work with Davinci as this model has been found to provide more sophisticated responses and higher-level outputs.

4.4. Other Research

Further research has been conducted in identifying and collecting a range of cyber incident response plans (CIRPs). These plans are being collected in PDF form and are being incorporated into the project by providing the OpenAI model with data / information to use in its responses.

We have decided that the CIRPs being collected are to have been published within a more recent timeframe (past 5 years) as this will provide our model with a more accurate and modern knowledge base.