EV-IRP Manager: An Electric Vehicle Incident Response Playbook Manager and Visualizer Toolkit

Kerem Alpdag∗, Naghmeh Moradpoor†, Ny Hasina Andriambelo‡, Paul Wooderson§, and Leandros Maglaras†
∗School of Computing and Mathematical Sciences, University of Greenwich, London, UK
†School of Computing, Engineering and the Built Environment, Edinburgh Napier University, Edinburgh, UK
‡School of Engineering Sciences and Techniques and Innovation, University of Antananarivo, Madagascar
§HORIBA MIRA Ltd, Nuneaton, UK

Abstract— **In the rapidly evolving realm of electric vehicle technology, safeguarding the cybersecurity of both electric vehicles and their charging infrastructure has become fundamental. The integration of electric vehicles and their charging stations into the broader grid introduces complex cybersecurity challenges, necessitating robust incident response strategies. Traditional cybersecurity playbooks often fall short in addressing the unique vulnerabilities associated with electric vehicles and their charging systems. The lack of publicly available community playbooks tailored to these needs leaves the electric vehicle ecosystem vulnerable to cyber threats that could compromise user privacy, vehicle functionality, and grid stability. In response to this, the project undertakes the creation of a foundational playbook for electric vehicle and electric vehicle charging station incident response, addressing a significant void in current cybersecurity practices. This paper introduces a Playbook Manager and Visualizer application, called EV-IRP, designed to enable users to upload, manage, and visualize electric vehicle and electric vehicle charging station incident response playbooks efficiently. Utilizing Python, Tkinter for GUI development, SQLite for database management, and Graphviz for visualization, the application facilitates a dynamic and responsive approach to maintaining up- to-date incident response strategies. The application is expected to streamline the management of incident response playbooks through procedure visualization, enhancing the cybersecurity posture of electric vehicle infrastructure. By converting textual playbook procedures into easily understandable diagrams, it facilitates a clearer understanding of response steps among users, thereby enhancing the overall efficiency and efficacy of incident response practices. Additionally, the research and development process, informed by a comprehensive literature review, contributes to the academic and practical understanding of cybersecurity best practices for electric vehicle technologies.**

Keywords— Electric Vehicle; Electric Vehicle Charging Systems; Incident Response; Cybersecurity Playbook; Visualization

I. INTRODUCTION

As the global automotive industry pivots towards sustain-ability and innovation, the adoption of Electric Vehicles (EVs) has accelerated, promising a greener future. However, this rapid integration of EVs and their charging infrastructure into the broader electrical grid has introduced an array of complex cybersecurity challenges. The digital and connected nature of these systems exposes them to potential cyber threats that could compromise user privacy, vehicle functionality, and even grid stability. Recognizing the urgency of this issue, this paper introduces a pioneering approach to enhancing the cybersecurity of EV ecosystems through the development of a foundational Playbook Manager and Visualizer application called EV-IRP. Currently, specialized cybersecurity measures are needed in the EV ecosystem. The vulnerabilities inherent in EVs and EV Charging Systems (EVCS) differ significantly from those of traditional IT environments, necessitating be-spoke cybersecurity solutions. Unlike conventional vehicles, EVs heavily rely on software, making them susceptible to software bugs, malicious hacking attempts, and unintended user errors. Moreover, EVCS, which serve as critical infrastructure nodes, handle sensitive data transactions and energy management tasks that are attractive targets for cybercriminals. The integration of these stations with home networks and larger grid systems further complicates their security land- scape, creating multiple points of potential exploitation.

Despite the critical nature of these vulnerabilities, there is a noticeable lack of publicly available cybersecurity community playbooks that address specific threats to EVs and their associated charging infrastructure. To the best of our knowledge, we have not found a single incident response playbook specifically addressing EVs and EVCS. Most existing cybersecurity frameworks are too generic to cover the specialized scenarios encountered in EV operations. This gap not only leaves the EV infrastructure vulnerable but also hampers the ability of stakeholders to respond effectively to incidents. Playbooks serve as valuable resources for learning and guidance, and their absence in this area is a significant drawback for enhancing cybersecurity. Essentially, having these playbooks would create a more secure, trusted, consistent, and instructive cybersecurity environment for EVs and EVCS.

In this paper we aim to fill a crucial gap by developing a Playbook Manager and Visualizer application called EV-IRP tailored specifically for EVs and EVCS. The application is designed to enable users, ranging from cybersecurity professionals to infrastructure managers, to upload, manage, and visualize specific incident response playbooks efficiently. By doing so, it addresses the acute need for an accessible, adaptable, and understandable tool that can assist in the swift

and effective mitigation of cyber threats within the EV domain.

The introduction of EV-IRP Manager and Visualizer stands to benefit the EV industry in multiple ways as follows:

- Standardization of Response Protocols: The application facilitates sharing and managing cybersecurity playbooks, promoting standardized response protocols industry-wide. This simplifies incident management across systems and teams while ensuring widespread adoption of best practices.
- Reduction in Incident Resolution Time: Visual tools aid in quicker comprehension and retention of information, reducing the cognitive load associated with complex procedures. Users can easily locate incidents using key- words or provided categories, expediting decision-making during cybersecurity incidents.
- Enhanced Training and Preparedness: The application serves as a valuable training tool for new cybersecurity personnel in the EV sector. Simulated incident scenarios depicted through the visualizer enhance learning and preparedness, creating a well-informed workforce capable of handling real-world threats.

Therefore, the objectives of this research paper are as follows:

- Conduct a literature review to understand cybersecurity challenges with EVs and charging infrastructure, identifying gaps for specialized playbooks.
- Design and implement a user-friendly Playbook Manager and Visualizer application called EV-IRP.
- Enable efficient management and categorization of cybersecurity incidents for quick access and systematic response.
- Develop visualization tools converting text-based play- book procedures into clear diagrams.
- Evaluate the application's impact on enhancing EV ecosystem cybersecurity.
- Propose future enhancements and research directions based on application development and evaluation findings.

The remainder of this paper is structured as follows: Section II reviews related work, Section III discusses the methodology, Section IV presents data collection and incident identification, Section V covers design, implementation, and results, Section VI addresses standards and compliance, and Section VII concludes with future work directions.

II. Literature Review

The literature reviewed underscores the importance of structured cybersecurity frameworks, highlighting the nuanced vulnerabilities and unique challenges EVs and EVCs technologies face. It explores seminal works discussing the effectiveness of cybersecurity playbooks, gaps in current practices, and technologies aimed at mitigating risks. This section lays the groundwork for developing our proposed EV-IRP Manager and Visualizer by examining both the inherent vulnerabilities in EVs and EVCs and the strategic responses advocated by current research.

The research highlighted in [1] underscores the vital role of standardized playbooks in improving cybersecurity responses, especially in complex environments with multiple decision- makers and systems. It advocates for structured incident response approaches to reduce variability and inefficiencies in decision-making. By proposing a playbook specification format, the paper promotes the synthesis of security knowledge into tailored procedures for specific risks and operational contexts. This standardization not only accelerates response times but also enhances decision quality, thereby strengthening overall security posture. Importantly, the research identifies a significant gap in publicly accessible playbooks tailored to specific sectors, which poses a critical vulnerability given the unique challenges and threat models faced by EVs and EVCs.

The study in [6] highlights the significance of incident response playbooks and the challenges in defining, using, and sharing them across different organizational contexts. These playbooks include crucial information on specific threats and organizational details for maintaining effectiveness and confidentiality. A key finding of the paper is the ambiguity in how organizations define their playbooks, impacting their usability across various cybersecurity actors who often need modifications for application. Importantly, there is a shortage of community playbooks tailored to specific sectors or technologies, such as EVs and EVCs.

In paper [2], the author presents a thorough survey of methods used to detect cybersecurity threats in smart vehicle systems, covering EVs and EVCs. Various techniques including intrusion detection systems (IDSs), anomaly detection, attack detection, and hybrid methods are explored, applied across in-vehicle networks (IVNs), inter-vehicle networks, ground vehicle power stations, and the Internet of Drones (IoD). A significant finding of this survey is the detailed analysis and comparison of different detection methods. This analysis includes a review of datasets, simulations, and key evaluation criteria, providing a comprehensive overview of current capabilities and shortcomings in this field. Addition- ally, the paper discusses challenges in detecting intrusions, anomalies, and attacks on smart vehicles, highlighting the need for ongoing research and development to enhance security measures as these technologies evolve. This underscores the importance of developing specialized security frameworks like our Playbook Manager and Visualizer, tailored specifically for EV and EVC ecosystems.

In paper [7], the authors delve into the structured method-ologies employed by U.S. federal agencies to tackle cybersecurity incidents and vulnerabilities. They focus on two main playbooks: the Incident Response Playbook and the Vulnerability Response Playbook, detailing procedures for effective cybersecurity management. The Incident Response Playbook is segmented into several phases: Preparation, Detection & Analysis, Containment, Eradication & Recovery, and Post- Incident Activities. It emphasizes the importance of preparation and continuous monitoring, utilizing techniques

like cyber threat intelligence (CTI) to proactively identify potential threats. The Vulnerability Response Playbook highlights the significance of addressing actively exploited vulnerabilities. It advocates for a systematic approach to identifying, evaluating, remediating, and reporting vulnerabilities. Standardized processes enable federal agencies to swiftly respond to critical vulnerabilities, enhancing overall security posture.

The papers [4] & [5] provide detailed insights into the cybersecurity challenges and risks associated with the increasing integration of EVs and EVCs. The first paper examines vulnerabilities in electric vehicle supply equipment (EVSE), detailing how these systems are susceptible to cyberattacks due to their connection with cloud services, EVs, and grid operators. It dis- cusses the severe impacts of such attacks, ranging from minor local disturbances to significant disruptions on a national scale. The study emphasizes adopting Information Technology (IT) and Operational Technology (OT) cybersecurity best practices to protect these infrastructures. The second paper explores systemic risks associated with the electrification of transportation, emphasizing broader implications for cybersecurity. It systematically examines cybersecurity vulnerabilities inherent in EV technologies and discusses the consequences of these threats, including the misuse of vehicle sensors and risks to personal and financial data. It identifies three main areas of sustainability that could be compromised due to cybersecurity threats: life and well-being, safe environment, and innovation and development. The increasing integration of digital technologies in EVs, while enhancing vehicle functionality and environmental sustainability, also escalates the risk profile, making them targets for cyberattacks.

Each source above contributes to understanding how tailored cybersecurity responses, like our proposed EV-IRP Manager and Visualizer, enhance security within the EV ecosystem, aligning with our paper's objective of standardized, effective cybersecurity solutions. For example, comprehensive cybersecurity approach presented in [2] aligns with our project's goal of standardizing and improving incident response protocols for EVs and EVCs, particularly in the absence of publicly accessible community playbooks addressing these threats. Our developed application could significantly contribute to addressing the current gap in EV cybersecurity practices. This absence poses a challenge due to the unique cybersecurity vulnerabilities inherent in EV and EVC systems, given their integration into the broader electrical grid and their digital, connected nature.

III. METHODOLOGY

The methodology employed in this project combines several advanced technological tools and programming frameworks to achieve its goals. Python, with its robust libraries and versatility, forms the backbone of the application's development, ensuring flexibility and powerful functionality. The graphical user interface (GUI) is crafted using Tkinter, facilitating user- friendly interaction and seamless navigation. SQLite is utilized for database management, providing a lightweight yet efficient system to store and retrieve data, while Graphviz supports the visualization aspect, transforming complex textual procedures into clear, understandable diagrams. This integration of technologies not only enhances the functionality of the Playbook Manager but also ensures that it is scalable and responsive to the evolving needs of the cybersecurity landscape.

By leveraging these technologies together, the project has succeeded in developing a user-friendly interface for managing cybersecurity incidents. The GUI developed using Tkinter ensures ease of use, catering to users with varying levels of technical expertise. Additionally, the integration of SQLite ensures efficient and secure handling of critical information related to incident titles, steps, and user credentials. This combination of tools makes the application an essential asset in the cybersecurity infrastructure, simplifying the complexities involved in incident response management.

Figure 1 represents the tools and technologies utilized in the creation of EV-IRP Manager. The application architecture comprises the following components, as depicted in Figure 2.

A. Database Layer

SQLite: Manages data storage and retrieval. The database schema encompasses tables for storing playbooks and user credentials, guaranteeing secure and efficient data management. The steps to develop the database layer include:

- Creating an SQLite database with tables dedicated to storing playbooks and user credentials.
- Ensuring the secure storage of user passwords by employing the bcrypt library for hashing.

B. GUI Layer

- Tkinter: Offers an interactive interface for users to engage with the application. This layer incorporates forms for adding, editing, and viewing incidents,

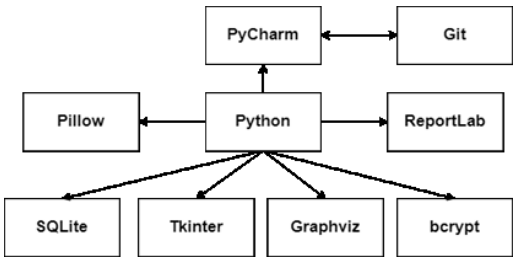

Fig. 1: Tool and Technology Stack

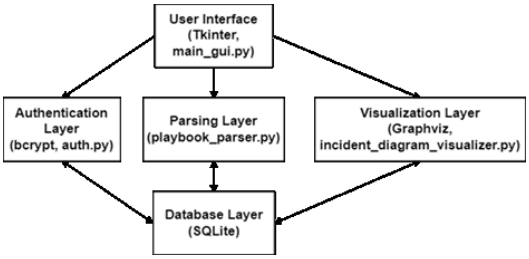

Fig. 2: System Architecture

alongside visualizing playbooks. The steps to develop the GUI layer include:

- Designing and implementing the main application window using Tkinter.
- Creating forms for uploading playbooks, adding, and editing incidents, and logging in.
- Developing interactive elements such as buttons, text fields, and list boxes for user interaction.

C. Visualization Layer

Graphviz: Utilizes this open-source graph visualization soft- ware to produce visual diagrams from text-based playbook data. This layer transforms procedural text into diagrams, facilitating users' comprehension and adherence to incident response steps. The steps to develop the visualization layer include:

- Integrating Graphviz to generate visual diagrams from the procedural text stored in the database.
- Developing functions to convert text-based procedures into graph format, ensuring clarity and ease of understanding.

D. Authentication Layer

auth.py: Implements user authentication functionalities. This module manages user login, password hashing using bcrypt, and secure storage of user credentials. It ensures that solely authorized users can access and modify sensitive data within the application. The steps to develop the authentication layer- include:

- Implementing user authentication using a simple login form.
- Ensuring secure storage and verification of user credentials with bcrypt.
- Adding access controls to restrict certain actions (e.g., adding or editing incidents) to authenticated users only.

E. Parsing Layer

playbook_parser.py: Responsible for parsing text-based playbooks. This module reads and processes playbook files, extracting pertinent information and storing it in the SQLite database. It supports the fundamental functionality of uploading and managing playbook data. The steps to develop the parsing layer include:

- Implementing functionality to parse text-based playbooks and store them in the SQLite database.
- Developing methods for adding, editing, and deleting playbook entries.
- Adding features to categorize and search incidents to facilitate quick access.

IV. Data Collection and Incident Idenfication

In developing the Playbook Manager and Visualizer for EV and EVC incident response, it was crucial to identify and document a comprehensive list of potential cybersecurity incidents. Our list comprises 30 distinct incidents. These incidents were identified, documented, and ranked through a comprehensive literature review, which included academic papers and industry reports, government and industry guidelines, technical surveys and reviews, as well as recent studies and articles, as follows.

A. Incident Sources and References

The incidents in the playbook were primarily drawn from the following sources:

1) Academic Papers and Industry Reports: Various academic papers and industry reports provide insights into the types of cybersecurity threats facing EV and EV charging infrastructure. Notable references include:

- The authors in [4] highlight vulnerabilities in EV charging systems and the potential impacts of cyberattacks, informing the categorization and ranking of incidents in the playbook.
- The authors in [5] discuss various emerging threats to EVs and their potential impact, providing a basis for several incidents in the playbook.
- The authors in [7] highlight the unique cyber vulnerabilities associated with EV battery packs, emphasizing the risks of overcharging and over-discharging, which can lead to battery degradation and safety hazards.
- The authors in [8] discuss a comprehensive framework for assessing vulnerabilities in EV charging systems, identifying various attack vectors such as man-in-the-middle attacks, denial-of-service attacks, and malware propagation.
- The authors in [9] provide an overview of cyber-physical threats to EV charging infrastructures and propose risk assessment methodologies, which informed the identification and categorization of incidents in our playbook.

2) Government and Industry Guidelines: Documents such as [3] provide a framework for categorizing and responding to cybersecurity incidents, which was adapted for EV-specific scenarios.

3) Technical Surveys and Reviews:

- The authors in [10] outline the communication challenges and security concerns in smart grid systems, relevant to EV charging networks.
- The authors in [11] provide insights into potential cyber- physical attacks on smart grids, including those involving EV charging stations.

4) Recent Studies and Articles: To ensure the incidents reflect the latest trends and threats in the EV domain, recent studies and articles were reviewed. Key references include:

- The authors in [12] provide an overview of cyber-physical threats to EV charging infrastructures and proposes risk assessment methodologies.
- The authors in [8] offer a detailed analysis of potential attack scenarios on EV charging infrastructures and their implications.

TABLE I: 30 EV & EVCS INCIDENTS

Incident	Name	Criticality
1	EV Charging Fraud via Vehicle Impersonation	Medium
2	Attacks Against Charging Networks	High
3	Charging Stations Attacking Multiple Vehicles	High
4	Remote Exploitation of EV Charging Stations	High
5	Inappropriate Content Displayed on EV Charging Station Screens	Low
6	Ransomware Attacks Against EV Charging Stations and Users	Critical
7	High Voltage Fault Codes Triggered by Hacking	Medium
8	MitM Attack on EV Charging Communication	Medium
9	Ransomware Attack on EV Charging Station Management System	Critical
10	Impersonation of Charging Station Admin Users	High
11	Disabling EV Charging Stations as Part of Cyberwar Efforts	Critical
12	Unauthorized Access to EV Charging Station	Medium
13	EV Charging Station Malware Infection	High
14	DoS Attack on EV Charging Network	High
15	Physical Tampering with EV Charging Hardware	High
16	Data Breach of Customer Information	Critical
17	Firmware Tampering in EV Charging Stations	High
18	Exploitation of Vulnerabilities in EV Charging Station Software	High
19	Securing EVs and Charging Stations with VSOCs	Medium
20	Physical and Remote Manipulation of EV Chargers	High
21	EV Battery Draining Attack via Compromised Charging Station	High
22	Overcharging Attack Leading to EV Battery Damage	High
23	Unauthorized EV Charging Due to Weak Authentication	Medium
24	Injection of False Data in EV Charging Station Telemetry	Medium
25	Theft of Service via EV Charging Station Network Penetration	Medium
26	Compromise of EV Charging Reservation System	High
27	Interference with Wireless Charging Communication	Medium
28	Mis-configuration of Charging Station Parameters	Medium
29	Supply Chain Attacks on EV Charging Station Components	High
30	Compromised Mobile Apps for EV Charging Stations	High

- The authors in [9] discuss the specific vulnerabilities and cybersecurity challenges of powertrain systems in modern electric vehicles.

B. *Incident Documentation and Rating*

Each incident in the playbook was documented with detailed descriptions, including preparation, detection, response, criticality, and criticality description. The criticality of each incident was ranked based on its potential impact on EV infrastructure, user safety, and operational continuity with the help of a literature review. An extensive review of current research helped identify common threats and their potential impacts. For example:

- Remote exploitation of EV charging stations and their potential for widespread network compromise [8].
- Physical tampering and its immediate risks to safety and operational disruption [7].
- Industry Best Practices: Best practices from government and industry guidelines were incorporated to ensure comprehensive coverage of incidents [3].

Table I represents the 30 incidents along with their criticality, and Table II represents the preparation, detection, and response for each incident. This information is gathered from the sources mentioned above.

V. DESIGN, IMPLEMENTATION & RESULTS

This section provides an in-depth exploration of the design, development process and technical details behind EV-IRP. It focuses on translating the analysis, design, and requirements into a robust software application. The primary objective is to present a comprehensive overview of the analysis & design, as well as implementation, highlighting the key components, functionalities, and technologies utilized.

A. *Analysis & Design*

The primary objective of the analysis and design phase is to gain a deep understanding of the system's functional requirements, non-functional requirements, hardware and soft- ware requirements, and platform requirements. To achieve this, we use UML diagrams such as use case, Figure 3 and sequence diagram, Figure 4. By employing these tools, we aim to provide a comprehensive and structured overview of the Playbook Visualizer application, facilitating its development across various desktop platforms. The use case diagram in Figure 3 shows what users can do within the application. It provides an overview of the functional requirements of the system and illustrates the interactions between the user and the application. The sequence diagrams depict the flow of messages and interactions between the system components to accomplish specific tasks, such as uploading a playbook, viewing and exporting an incident diagram, adding an incident, and editing an incident. The sequence diagram, depicted in Figure 4, shows the process of viewing and exporting a diagram. First, the user selects an incident, and the application highlights that incident. Then, PlaybookVisualizerApp calls the view_incident_details() method to invoke the IncidentDiagramVisualizer class. After that, the class calls the necessary methods and displays the diagram to the user. The user can then click on the export button to export the diagram as PNG or PDF.

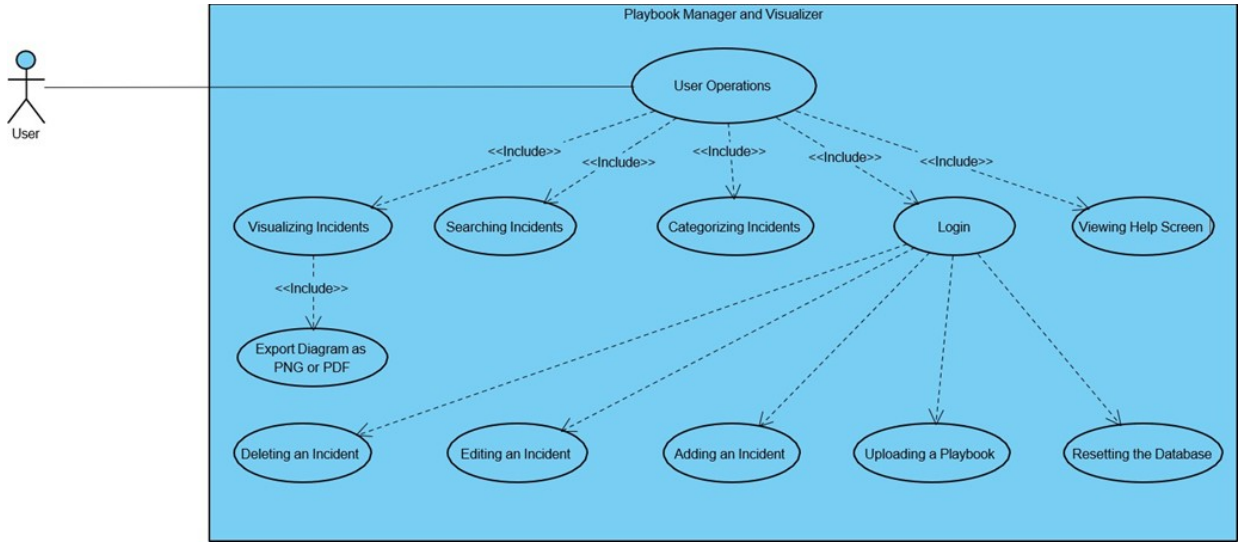

Fig. 3: Use Case Diagram for EV-IRP

After the diagram is exported, the IncidentDiagramVisualizer is destroyed, and an export message is shown to the user.

B. Implementation

EV-IRP is developed using Python, a versatile and widely adopted programming language known for its simplicity and extensive libraries. The choice of Python enables seamless execution of the application across different desktop platforms, including Windows, macOS, and Linux. Furthermore, the implementation of the Playbook Visualizer relies on specific libraries and frameworks, enhancing its capabilities in database management, graphical user interface (GUI) development, and diagram generation. Now, let's proceed to the specific implementation details, examining each aspect of the Playbook Visualizer's development and uncovering the unique features and functionalities that make it a valuable tool in the domain of incident management.

C. System Architecture

The Playbook Visualizer is implemented using a modular and object-oriented architecture. It consists of several key classes that work together to provide the desired functionality. The following classes play significant roles in the system's architecture:

- PlaybookVisualizerApp: This class serves as the main application class that initializes and manages the user interface, handles user interactions, and coordinates with other components. It includes attributes for the Playbook-Parser, Authenticator, and IncidentDiagramVisualizer instances, as well as the root Tkinter window. The class methods cover a wide range of functionalities, including opening forms, managing incidents, handling the database connection, and uploading playbooks.
- AddIncidentForm: This class provides a form for adding new incidents to the playbook. It includes methods for initializing the form, setting up the user interface, retrieving the next incident number from the database, handling form submission, and inserting a new procedure into the database.
- EditIncidentForm: This class provides a form for editing existing incidents in the playbook. It includes methods for initializing the form, setting up the user interface, retrieving the incident details, and handling the form submission for editing an incident.
- HelpScreen: This class provides a help screen with information about the application. It includes methods for initializing the help screen and setting up the user interface.
- LoginForm: This class provides a login form for authenticating users. It includes methods for initializing the login form, setting up the user interface, and handling the login process.
- IncidentDiagramVisualizer: This class generates and displays diagrams for incidents. It includes methods for initializing the visualizer, generating a diagram based on incident details, adding the title and category to the diagram, displaying the diagram, and exporting the diagram as a PNG or PDF file.
- PlaybookParser: This class parses playbook files and categorizes incidents. It includes methods for initializing the parser, categorizing incidents based on their title, inserting parsed incidents into the database, and reading and interpreting the playbook file.
- Authenticator: This class handles user authentication and password management. It includes methods for initializing the authenticator, hashing passwords, verifying pass- words against stored hashes, retrieving stored password hashes, and authenticating users.
- Database: This class/script manages the creation and initialization of the database. It includes the method to

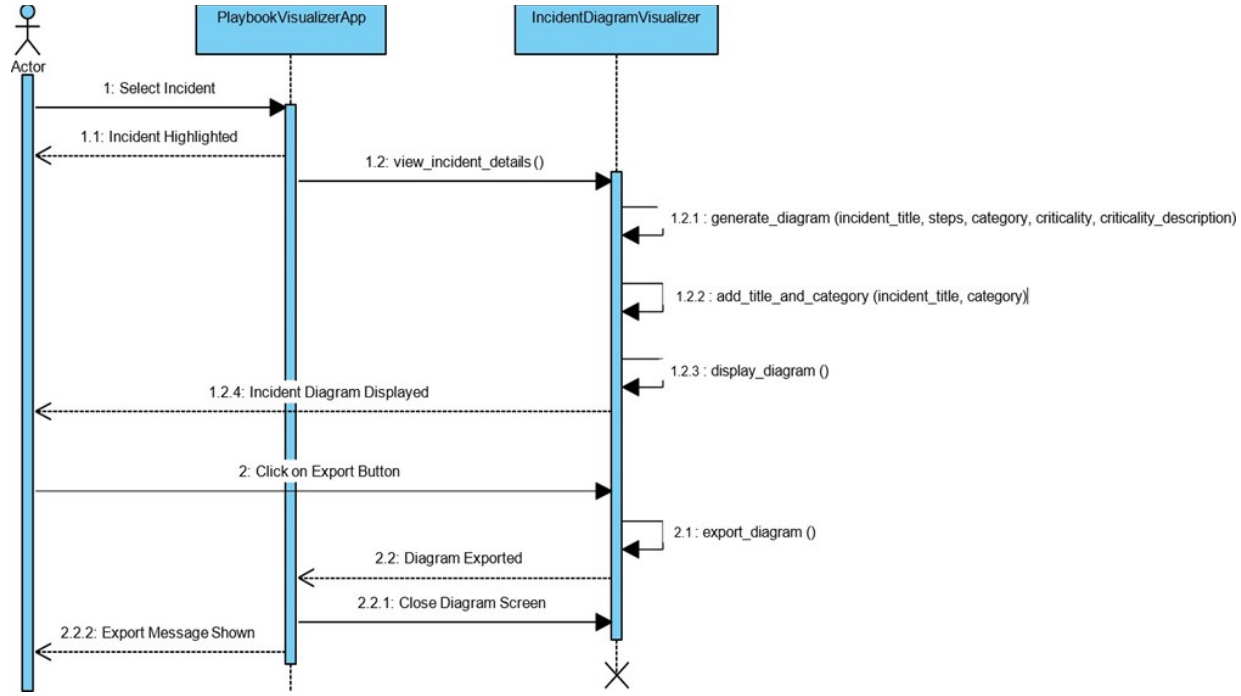

Fig. 4: Sequence Diagram for Viewing and Exporting an Incident Diagram

create database, which creates the database schema and initializes the database.

The system architecture follows a layered approach, with separate classes for user interface components (such as AddIncidentForm, EditIncidentForm, LoginForm, HelpScreen), core functionality (such as PlaybookParser, Authenticator, Incident- DiagramVisualizer), and databasemanagement (Database). This modular design allows for easier maintenance, testing, and future enhancements. By following this modular and object-oriented architecture, the Playbook Visualizer achieves a clear separation of concerns, promotes code reusability, and facilitates future scalability and extensibility.

D. Main Application Screen

The main application screen serves as the central interface for users to manage incidents, view diagrams, and access other functionalities. Upon launching the application, users are presented with an incident list, categorized views, search functionality, and options to add, edit, or delete incidents. The main screen also provides buttons to upload playbooks, reset the database, and access the help screen, Figure 5. The UI component leverages Tkinter's widgets, such as Listbox, Entry, Button, and Menu, to create an intuitive and functional interface.

- Add Incident Form: The Add Incident Form allows users to add new incidents to the playbook. It includes fields for the incident name, preparation, detection, response, criticality, and criticality description.The form is designed to ensure that all required information is captured accurately. It utilizes Tkinter's Entry and Combobox widgets to create a user-friendly interface for inputting incident details.
- Edit Incident Form: The Edit Incident Form allows users to modify existing incidents. It pre-populates the form fields with the current details of the selected incident, allowing users to make necessary changes. The form includes fields for the incident name, preparation, detection, response, criticality, and criticality description. Similar to the Add Incident Form, it utilizes Tkinter's Entry and Combobox widgets to facilitate user input.
- Login Form: The Login Form provides a secure way for users to authenticate themselves before performing certain actions. It includes fields for the username and password and uses the Authenticator class to verify user credentials. The form is implemented using Tkinter's Entry widget for secure password input.
- Help Screen: The Help Screen offers users guidance on how to use the application, including details about the document format that the application accepts. It provides a clear and concise explanation of the application's features and usage. The help screen is implemented using Tkinter's Label widget to display the help text.

- Incident Diagram Screen: The Incident Diagram Screen displays the generated incident diagram. It allows users to visualize the incident workflow and provides options to export the diagram as a PNG or PDF file. The screen is implemented using Tkinter's Label and Button widgets to display the diagram and provide export functionality. Figure 6 to Figure 9 depict four examples of incidents with varying levels of criticality (Low, Medium, High, Critical). The steps to generate each incident diagram have been taken from Table II, which is based on our comprehensive literature review of EV and EVCS cyber-security incidents.

By implementing these user interface components using Python and Tkinter, the Playbook Visualizer ensures a seamless and intuitive user experience, allowing users to effectively manage incidents and visualize incident workflows.

E. Libraries Utilized in Development

The development of the Playbook Visualizer leveraged several Python libraries to implement various functionalities. These libraries were chosen for their robustness, ease of use, and ability to integrate seamlessly into the application. This section discusses the key libraries used in the development process, explaining their roles and contributions to the application.

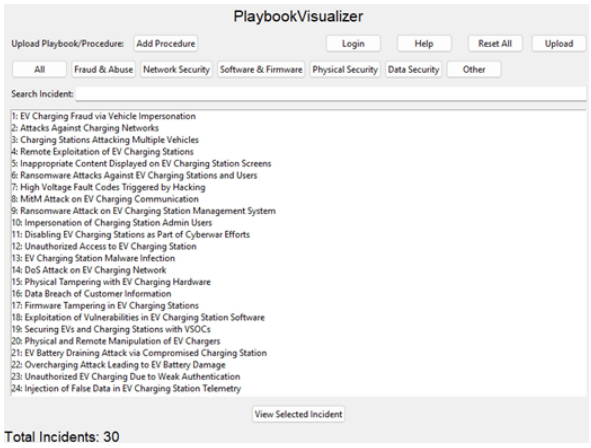

Fig. 5: Main Screen of the Application

1) Tkinter: it is the standard GUI library for Python. It provides a powerful object-oriented interface to the Tk GUI toolkit. Usage in Playbook Visualizer:

- User Interface Components: Tkinter was used extensively to create the main application window, forms, buttons, labels, and other UI elements. It facilitated the creation of a responsive and user-friendly interface.
- Event Handling: Tkinter's event handling mechanisms were used to manage user interactions, such as button clicks, form submissions, and context menu actions.

SQLite3: it is a library that provides a lightweight, disk-based database. It is self-contained, serverless, and requires no configuration. Usage in Playbook:

- Database Management: SQLite3 was used to create and manage the application's database. It stored

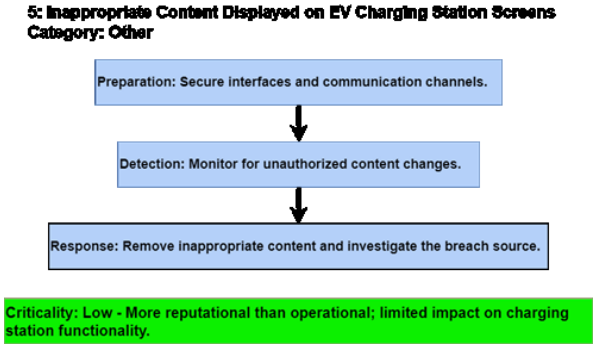

Fig. 6: Incident Diagram Example (Low Criticality)

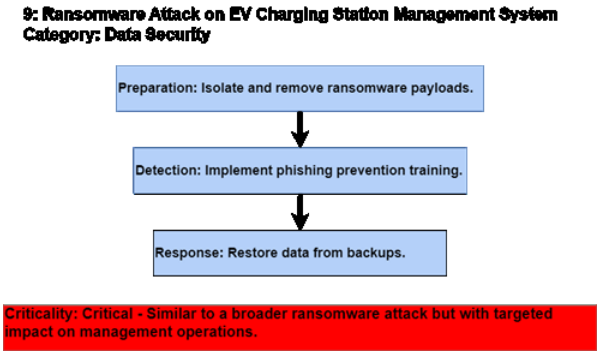

Fig. 7: Incident Diagram Example (Critical Criticality)

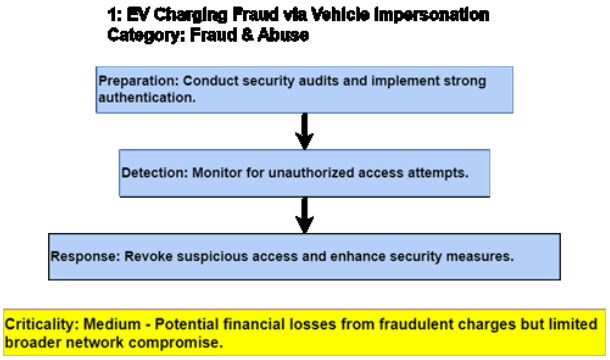

Fig. 8: Incident Diagram Example (Medium Criticality)

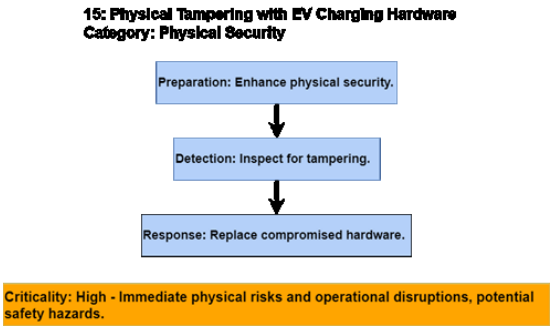

Fig. 9: Incident Diagram Example (High Criticality)

TABLE II: 30 EV & EVCS INCIDENTS: PREPARATION, DETECTION & RESPONSE (REFER TO TABLE I FOR INCIDENT DETAILS)

Incident	Preparation	Detection	Response
1	Conduct security audits, implement strong authentication	Monitor for unauthorized access attempts	Revoke suspicious access, enhance security measures
2	Deploy network segmentation	Analyse network traffic for anomalies	Isolate compromised segments, mitigate threats
3	Update charging stations software	Use IDS to detect malicious activities	Patch vulnerabilities, disable affected stations
4	Implement hybrid intrusion detection, encrypt communications	Identify signs of remote exploitation	Secure communication channels, block exploit paths
5	Secure interfaces, communication channels	Monitor for unauthorized content changes	Remove inappropriate content, investigate the breached source
6	Establish backup strategies	Identify ransomware infections	Restore data from backups, educate users on phishing prevention
7	Implement strict input validation on software	Monitor for unexpected fault codes	Conduct penetration testing, remedy identified vulnerabilities
8	Implement end-to-end encryption	Educate users on secure connections	Use secure communication protocols
9	Isolate and remove ransomware payloads	Implement phishing prevention training	Restore data from backups
10	Implement RBAC and authentication protocols	Regularly review access permissions	Update access controls, revoke unauthorized access
11	Develop a BCP	Collaborate for threat intelligence sharing	Coordinate responses with partners
12	Implement multi-factor authentication	Conduct security audits	Revoke unauthorized access
13	Isolate infected stations	Run malware scans	Restore systems from backups
14	Implement rate limiting	Monitor for traffic anomalies	Redistribute network traffic
15	Enhance physical security	Inspect for tampering	Replace compromised hardware
16	Encrypt sensitive data	Identify data breach	Notify affected users, reset passwords
17	Secure firmware update mechanisms	Verify firmware integrity	Reinstall trusted firmware
18	Apply security patches	Conduct vulnerability assessments	Monitor, mitigate suspicious activitie
19	Establish a centralized VSOC	Integrate advanced threat detection tools	Address incidents based on VSOC insights
20	Enhance physical security measures	Inspect for tampering signs	Replace compromised hardware, secure communication
21	Monitor energy usage patterns	Identify anomalies in energy usage	Disable remote access, update firmware
22	Implement strict charging controls	Monitor for irregular charging behavior	Disable affected stations for inspection
23	Strengthen authentication mechanisms	Audit charging sessions	Update access credentials regularly
24	Validate data integrity	Monitor for unexpected data patterns	Secure communication channels
25	Secure network endpoints	Scan for unauthorized access points	Implement network segmentation
26	Secure backup of reservation system	Monitor for unauthorized access	Restore system, update security measures
27	Ensure robust communication protocols, shielding against interference	Monitor for communication disruptions or anomalies in wireless charging sessions	Investigate, mitigate sources of interference, update communication protocols
28	Regularly audit, validate configuration settings	Monitor for deviations from standard operating parameters	Correct mis-configurations, review change management processes
29	Vet suppliers, implement rigorous quality control measures	Inspect components for anomalies or inconsistencies	Replace compromised components, enhance supply chain security
30	Implement secure coding practices, regular app security reviews	Monitor app stores for malicious versions, monitor user reports of anomalies	Revoke malicious apps, notify users to update or reinstall secure versions

incident details, user credentials, and other relevant data.

- CRUD Operations: The library facilitated Create, Read, Update, and Delete (CRUD) operations on the database, enabling efficient data management.

2) *bcrypt:* it is a password hashing library designed for secure password storage. It is based on the Blowfish cipher and includes features to protect against rainbow table attacks. Usage in Playbook Visualizer: Password Hashing: bcrypt was used to hash user passwords securely before storing them in the database.

- Password Verification: The library's verification functions ensured that user-provided passwords matched the stored hashed passwords during the authentication process.

3) *Graphviz:* it is an open-source graph visualization software It provides a way of representing structural informa- tion as diagrams of abstract graphs and networks. Usage in Playbook Visualizer:

- Diagram Generation: Graphviz was used to generate visual diagrams of incident workflows. It created nodes and edges based on incident steps, providing a clear visual representation.
- Customization: The library allowed customization of node shapes, colors, and styles, enhancing the clarity and aesthetics of the diagrams.

4) *Pillow:* it is a Python Imaging Library (PIL) fork. It adds image processing capabilities to Python, allowing for opening, manipulating, and saving many

different image file formats. Usage in Playbook Visualizer:

- Image Manipulation: Pillow was used to open, edit, and save images. It added titles and categories to the diagrams generated by Graphviz.
- Image Export: The library facilitated exporting diagrams as PNG files, ensuring high-quality image outputs.

5) *ReportLab:* it is a robust library for creating PDF documents in Python. It allows for complex layouts and dynamic content generation. Usage in Playbook Visualizer:

- PDF Export: ReportLab was used to export incident diagrams as PDF files. It managed the layout, scaling,and placement of images on the PDF pages, ensuring that diagrams were accurately represented.

By leveraging these libraries, EV-IRP was able to provide a comprehensive set of features, from a robust user interface to secure authentication, efficient data management, and clear visual representations of incident workflows. Each library contributed specific functionalities that enhanced the overall capability and performance of the application. Overall, the implementation of the EV-IRP has successfully achieved the project's objectives. The application provides an intuitive user interface, robust incident management functionality, and clear visual representations of incident workflows.

VI. Standards and Compliance Requirements

Cybersecurity is now regulated in the automotive industry with the introduction of UN Regulation 155, adopted by many countries since 2022, including the EU, Japan, Korea, and Australia. This regulation requires vehicle manufacturers to implement a cybersecurity management system and meet specific requirements for type approval. China and other regions are also introducing similar regulations. UN Regulation 155 mandates that manufacturers detect and respond to emerging threats and vulnerabilities throughout a vehicle's lifetime, and implement timely responses to incidents. Annex 5 lists typical threats, including manipulation of remotely operated systems like the charging pile. To support compliance, ISO/SAE 21434 was published in 2021, providing a framework for cybersecurity engineering activities, including monitoring, vulnerability management, and incident response. This standard requires incident response plans to ensure vulnerabilities are addressed. These regulations drive the need for effective incident response solutions, such as the EV-IRP, which offer practical tools for managing EV incidents. These playbooks can also be used by other organizations, like charge point operators and fleet charging facility operators.

VII. Conclusion and Future Work

In conclusion, EV-IRP successfully addresses the need for an intuitive and efficient tool for managing cybersecurity incidents in EVs and EVCS. The development and implementation of this tool have demonstrated its capability to handle essential functionalities such as adding, editing, viewing, and deleting incidents, as well as generating and exporting visual diagrams of incident workflows. The application also provides a user- friendly interface. While EV-IRP has proven its utility, there is always room for enhancement and expansion. The following are key areas where the tool can be further improved:

- Real-time Collaboration: Introducing real-time collaboration features would allow multiple users to work on the same playbook simultaneously.
- Integration with External Systems: Expanding the tool's integration capabilities with other cybersecurity tools, incident management systems, and data sources can enhance its functionality.
- Developing mobile and web versions of the EV-IRP: This will enhance accessibility for users managing incidents on the go, requiring responsive design and optimized performance for various devices and platforms.
- Advanced Reporting and Analytics: Adding features for generating detailed reports based on incident data. Users could benefit from customizable reports, trend analysis, and visual analytics dashboards to gain deeper insights into their incident management processes.

References

[1] Applebaum, A. et al. (2018) 'Playbook oriented cyber response', 2018 National Cyber Summit (NCS) [Preprint]. doi:10.1109/ncs.2018.00007.

[2] Banafshehvaragh, S.T. and Rahmani, A.M. (2023) 'Intrusion, anomaly, and attack detection in smart vehicles', Microprocessors and Microsystems, 96, p. 104726. doi:10.1016/j.micpro.2022.104726.

[3] TISZA, O. C. (2022). Federal Government Cybersecurity Incident & Vulnerability Response Playbooks.

[4] Johnson, J. et al. (2022) 'Review of Electric Vehicle Charger Cybersecurity vulnerabilities, potential impacts, and defenses', Energies, 15(11),

[5] p. 3931. doi:10.3390/en15113931.

[6] Muhammad, Z. et al. (2023) 'Emerging cybersecurity and privacy threats to electric vehicles and their impact on human and environmental sustainability', Energies, 16(3), p. 1113. doi:10.3390/en16031113.Schlette, D., Empl, P. and Caselli, M. (2023) Do You Play It by the Books? A Study on Incident Response Playbooks and Influencing Factors. doi:10.1109/SP54263.2024.00060.

[7] Sripad, Shashank & Kulandaivel, Sekar & Pande, Vikram & Sekar, Vyas & Viswanathan, Venkatasubramanian. (2017). Vulnerabilities of Electric Vehicle Battery Packs to Cyberattacks on Auxiliary Components

[8] Reeh, D., Tapia, F. C., Chung, Y. W., Khaki, B., Chu, C., & Gadh, R. (2019, June). Vulnerability analysis and risk assessment of EV charging system under cyber-physical threats. In 2019 IEEE Transportation Electrification Conference and Expo (ITEC) (pp. 1-6). IEEE

[9] Ye, J. et al. (2021) 'Cyber–physical security of powertrain systems in modern electric vehicles: Vulnerabilities, Challenges, and future visions', IEEE Journal of Emerging and Selected Topics in Power Electronics, 9(4), pp. 4639–4657. doi:10.1109/jestpe.2020.3045667.

[10] Yan, Y., Qian, Y., Sharif, H., & Tipper, D. (2012). A survey on smart grid communication infrastructures: Motivations, requirements and challenges. IEEE communications surveys & tutorials, 15(1), 5-20.

[11] He, H., & Yan, J. (2016). Cyber-physical attacks and defences in the smart grid: a survey. IET Cyber-Physical Systems: Theory & Applications, 1(1), 13-27.

[12] Acharya, S., Dvorkin, Y., Pandz˘ic´, H., & Karri, R. (2020). Cybersecurity of smart electric vehicle charging: A power grid perspective. IEEE access, 8, 214434-214453