International Journal of Information Security (2026) 25:151
https://doi.org/10.1007/s10207-026-01263-9

REGULAR CONTRIBUTION

Generating semi-automated security playbooks for vulnerability mitigation from unstructured advisory data

Daniel Oberhofer[1] · Johannes Grill[2] · Günther Pernul[2] · Stefan Schönig[1]

Received: 30 August 2025 / Accepted: 15 April 2026

Abstract

Automated security risk management addresses threats by implementing mitigation and remediation strategies described in public security advisories. These advisories, regularly published by independent Cyber Emergency Response Teams (CERTs), are typically presented as unstructured text and supplemented with additional security-relevant information sources. Using a Large Language Model (LLM) orchestration framework, we generate security playbooks in a standardized meta-model, visualize them with Business Process Modeling and Notation (BPMN) diagrams, and classify them into actionable tasks. We present the theoretical foundations of the framework, describe the underlying meta-model, and conduct an experimental study that produces a dataset of 725 security playbooks. Our subsequent content analysis shows that advisory-based playbooks focus on update tasks, but also include defensive themes like disabling vulnerable policies or restricting network access.

Keywords Security Advisories · Playbooks · LLM · BPMN · Security Management

1 Introduction

Attackers have developed increasingly complex and extensive cyberattacks, which lead to an increasing number of software and hardware vulnerabilities listed in the National Vulnerability Database (NVD) [1, 2]. To address this increase in vulnerabilities and security threats, organizations must implement and continuously enhance their information security management systems [3]. This ongoing effort forms a critical part of a comprehensive security risk management lifecycle, which includes assessing risks, responding to and mitigating risks, and continuously monitoring them [4]. Industrial systems, while being historically reliant on proprietary technologies and isolated from the internet, are increasingly being connected and integrated with traditional IT systems [5]. This growing interconnection introduces new vulnerabilities and significantly raises the potential impact of security incidents [6]. A security risk management system is essential for all organizations, but particularly critical for industrial systems, as these incidents also possess the risk of endangering the physical world [7]. Such systems must also integrate external sources of threat intelligence, beyond internal security knowledge, to adequately address the evolving threat landscape, thereby making security risk management more effective and ensuring that important vulnerabilities and mitigation strategies are not overlooked [8, 9]. In this context, organizations consult security advisories that describe vulnerability mitigation, observed attack behavior, trending threat groups, and defensive best practices to address emerging threats to their systems [10]. Cyber Emergency Response Teams (CERTs) publish advisories as part of their services to different organizations in the public [11] or private sector [12, 13]. Although product vendors publish advisories for their product portfolios, CERTs cover a broader spectrum of security topics. As CERTs have responsibilities in a diverse set of security processes, ranging from proactive security monitoring to reactive incident response to general and advice-based security quality management [14, 15], they often enrich their advisories with additional information that

✉ Daniel Oberhofer
daniel.oberhofer@ur.de

Johannes Grill
johannes.grill@ur.de

Günther Pernul
guenther.pernul@ur.de

Stefan Schönig
stefan.schoenig@ur.de

[1] Chair of Process-based Information Systems, University of Regensburg, Bavaria, Germany

[2] Chair of Information Systems, University of Regensburg, Bavaria, Germany

Published online: 01 September 2026

goes beyond vulnerability mitigation. Even though efforts at format standardization, e.g., the Common Security Advisory Framework (CSAF) [16], exist, such complex security advisories are often published in the form of unstructured text on the CERTs' platforms [17]. Such unstructured and information-rich advisories lead to an organization's security analyst investing an extensive amount of manual workload in extracting the security process workflow, which is not feasible when facing the increase in the amount and diversity of relevant security advisories [18]. So has the amount of published security advisories, e.g., in the EU-CERT, doubled over the last years, from 2020 to 2024 [19]. At the same time, complying with regulations and demonstrating such compliance has become increasingly important in cybersecurity [20]. Implementing such advisories is a central part of organizational security risk management, and the underlying security processes are often mandatory, as mandated by government legislation and regulations, such as the NIS2 directive [21].

While transforming from unstructured to structured processes presents a challenge in itself [22], security analysts must implement complex advisory processes that can include multiple dependencies and technical steps [10]. The concept of cybersecurity playbooks addresses these implementation challenges, as it helps with structuring information, enables inter-organizational information sharing, and promotes scalability through process automation [22]. Security playbooks are structured security processes in various data formats, such as the *Collaborative Automated Course of Action Operations* (CACAO) or *Business Process Model and Notation* (BPMN) [23]. These processes can also be enriched with system-specific and command-and-control specifications, such as OpenC2 [24], allowing for the automated or semi-automated execution of process flows. Although full playbook automation may not be feasible, as playbooks must be enriched with context-related data, generating playbooks from broader CERT advisories, structuring and visualizing them for security experts, and automating selected steps can enhance the integration of security advisories into organizational security processes and improve overall risk management [25]. Especially, as Security Risk Management on an information system level can be broken down into the main steps of *Risk Assessment*, *Risk Response* and *Risk Monitoring*, these steps are often connected through complex checklists or reports [4]. Information exchange in between the steps of assessing risks, e.g. in form of evaluating mitigation strategies within advisories, and responding to risks, e.g. by implementing targeted response playbooks, is often represented by manual and complex sub-tasks. Automating this connection in between assessment and response, would help organizations to take a step toward automating and accelerating the risk mangement process. In addition to providing concrete automation solutions, understanding which security themes and defense strategies are most frequently represented in playbooks derived from advisories provides valuable insights for both researchers and practitioners. An analysis of advisories highlights recurring focus areas and helps identify potential gaps where current advisories provide limited guidance.

We use an architecture that orchestrates a Large Language Model (LLM) to extract security playbook workflows out of unstructured security advisories, visualizes them in BPMN, classifies tasks based on their automation capabilities, and analyzes the playbooks for common defensive themes and strategies. We demonstrate our approach by generating a dataset of 725 security playbooks from three different CERTs and evaluate the quality of the generated dataset in a two-fold way. This work discusses the following research questions (RQs):

RQ1: *How can unstructured information from cybersecurity advisories be transformed into structured security playbooks?*

RQ2: *Which security themes and defense strategies are most commonly represented in playbooks derived from advisories?*

While answering the RQs, we provide the following contributions: (1) Description of the theoretical concept for the generation of the playbooks, which also defines the components of security advisory-based playbooks, within a meta-model. (2) Description of an implemented architecture stack for the generation of security advisory-based playbooks that combines the threat-based perspective of security advisories with automated vulnerability mitigation tactics. We further published our code open-source [1]. (3) Generation and analysis of an open-source published dataset of 725 semi-automated security playbooks. As the results of our evaluation are based on real-world data, our dataset can be challenged and used within the community for data analysis purposes. The paper is structured as follows:

In Section 2, a preliminary background about relevant concepts is given. In Section 3, the body of related work is presented. In Section 4, we provide an overview of the orchestration concept, including a proposal for a corresponding software architecture. In Section 5, we present the evaluation approach of our method, which includes an experimental study for generating playbooks, accompanied by a thorough quality assessment. In Section 6, we present the results of the experiment evaluation and describe the common themes found within the playbooks. In Section 7, we analyze the results in depth, providing reflections on the dataset and discussing the practical implications for CERTs and orga-

[1] https://github.com/security-ratisbona/Sec_Advisory_BPMN_Generator

nizations. Section 8 briefly summarizes and concludes the paper.

2 Background

2.1 Cybersecurity Advisories

Technology and system vendors, as well as external security organizations (e.g., CERTs), regularly publish security advisories that address existing software or hardware vulnerabilities. Implementing such advisories in response to discovered vulnerabilities can be considered industry practice. These vulnerabilities are described in open data formats, such as *Common Vulnerabilities and Exposures (CVE)*, and published in open-source databases, e.g., the NVD [26] or the European Union Vulnerability Database (EUVD) [27]. To minimize their security risk and counter daunting threats to their systems, organizations must identify their vulnerabilities, monitor platforms for relevant security advisories, and adapt the included steps into actual security measures [10]. There are efforts of standardization, in the form of the semi-structured CSAF format [16], which is mainly used by major vendors like Siemens or the Cybersecurity & Infrastructure Security Agency (CISA), but lacks a general adoption in practice [28]. While CSAF provides detailed descriptions of vulnerabilities along with targeted mitigation guidance for affected products, it does not specify a sequential order of tasks, incorporate conditional workflow logic, or provide executable scripts or precise modifications of system properties. Monitoring advisory databases, identifying relevant workflows, extracting mitigation strategies, and finally implementing them is a manual and extensive task for systems administrators and security experts [17, 18]. Especially the unstructured advisories, published by CERTs, obscure valuable information behind long paragraphs of text, while also enriching them with additional information, e.g., found in [19, 29, 30]. Such advisories include opaque security processes that go beyond simply patching or mitigating software vulnerabilities.

2.2 Cybersecurity Playbooks

During the evolution from traditional business process management systems to security specific Security Orchestration, Automation, and Response (SOAR) platforms, the concept of playbooks became an integral part of security research [31]. While security playbooks are most known for automating incident response processes within SOAR platforms [13], they can be used as a support for arbitrary organizational processes [23]. In the context of cybersecurity, they are always linked to a specific threat or incident (e.g., advisories), they can be represented as a structured process with distinct steps, and they always assist humans with their technology and data interactions [23]. Playbooks can be further divided into highly specific playbooks tailored to an organizational environment [22] and more generic, community-centered playbooks that describe general use cases and process steps at a high level [23]. Playbooks have achieved full integration into cybersecurity practices through format standardization [32], such as CACAO [33] or BPMN, which has improved the sharing possibilities of playbooks [34]. While the generic community playbooks benefit from increased sharing potential, based on exchangeable data formats, visualization, and understandable use cases, the playbooks specific to organizations require enrichment with context information, e.g., IP addresses, and therefore allow for complete process automation [22].

2.3 OpenC2

OpenC2 is a language specification designed to enhance the automation of security processes by providing standardized commands for interacting with security systems and individual components [24]. An OpenC2 command consists of a standardized action and a targeted consumer that interprets and executes the action, returning a corresponding response. This abstraction enables OpenC2 commands to be vendor-agnostic, allowing for seamless communication across different security systems, regardless of the manufacturer. In OpenC2 version 1.0, a total of 20 actions are defined, including update, start, and query [24]. However, OpenC2 currently provides only limited support for generic commands across a small set of actions. Therefore, we do not use OpenC2 to create generic commands but rather to classify the extracted playbook steps based on the defined actions, enabling more effective content analysis.

2.4 LLM Orchestration

LLM orchestration refers to the structured coordination of interactions between LLMs and external components, such as data sources, tools, and predefined workflows, to achieve complex tasks. Rather than relying on a single prompt and response, orchestration leverages a combination of techniques to guide the model's reasoning and output toward a specific goal, thereby streamlining the development of LLM-based applications [35, 36]. Key concepts include *prompt engineering*, where carefully crafted instructions orchestrate the model's behavior, and *few-shot prompting*, which demonstrates the desired output through illustrative examples. Additionally, orchestration often involves *output formatting* to ensure that generated results conform to predefined schemas or structured representations, making them easier to integrate into downstream processes.

We use LLM orchestration to derive structured cybersecurity playbooks from unstructured advisory texts. By providing the advisory content, a target output format, and representative examples, the LLM generates playbooks that define workflows and actionable tasks in a human and machine-readable format.

3 Related Work

Our method contributes to prior work in four aspects. First we build the understanding of playbook generation and automation (see Sec. 3.1), second we contribute to using BPMN as a tool for security purposes (see Sec. 3.2), third we contribute to research on extracting security process workflows out of unstructured security information (see Sec. 3.3). At last, we contribute to analyzing trends and themes in information sources based on vulnerabilities (see Sec. 3.4).

3.1 Generating and Automating Playbooks

Generation. Research on playbooks focuses on the efficient generation of playbooks, based on various data sources in diverse security contexts. In [25], security advisories are also used, but their word-stemming and feature-matching approach yields low accuracy on unstructured text and still relies on converting advisories into CSAF. Unlike their method, which extracts isolated steps, our approach builds complete vulnerability mitigation workflows. In [37], fine-tuned LLMs transform incident documentation into structured mitigation strategies in the CACAO format. Their framework incorporates threat intelligence for incident response, whereas our work focuses on vulnerability mitigation. In [38], Cyber Threat Intelligence (CTI) reports are used to infer playbook steps tailored to post incident recovery activities. Their model uses semantic prompt engineering to extract steps, but focuses on identifying isolated steps, which is founded on the different use case of using CTI reports, rather than advisories, where we can extract complex workflows. In [39] and [40], their graph-based approach does not include finished mitigation strategies, but instead generates responses by matching attack tactics and techniques with corresponding countermeasures. In contrast, their approaches extend simple playbooks with external knowledge or derive new ones by combining multiple data sources, rather than generating complete playbook workflows from mitigation strategies. Similarly, [41] incorporate web-based information on incident response steps to support the manual generation and sharing of playbooks by human experts. In [42], human experts are combined with a deep learning–based recommender system that uses live incident data to extend baseline mitigation strategies in high-level SOAR playbooks. While effective for time-critical SOAR environments, our approach instead focuses on fully automated, time-insensitive playbook generation from vulnerability data. More broadly, both [42] and [43] integrate human experts into the playbook generation lifecycle, whereas we limit their role to semi-automated execution. Others, such as [44] and [45], focus on establishing different structuring or standardization templates that aim to improve the quality of the generated playbooks, rather than developing new generation methods. Overall, prior work on playbook generation has relied mainly on pre-structured data sources, targeted different application domains, and overlooked the potential of extracting complex workflows based on novel data sources.

Automation. While prior research mainly focused on improving the generation of playbooks, it only acknowledges automated playbook execution as relevant, but provides no specific solutions. Empl et al. also recognize OpenC2 as relevant for a standardized playbook execution, but use it only to derive classes for their playbook steps [25]. In addition, they acknowledge the necessity of human oversight in interactions with critical infrastructure and consequently advocate for semi-automated approaches in this context. Our perspective differs in that we consider human oversight essential during the execution of playbooks, rather than in their generation. In [22], they integrate the automation capabilities of BPMN into their playbook sharing framework. Although they share our position that playbook automation is necessary, they do not propose detailed solutions in this regard. We also recognize automated playbook execution as relevant for research, but still rely on a semi-automated approach.

3.2 BPMN and Playbooks

Research on BPMN in the context of cybersecurity is mainly conducted on the modeling of security aspects into existing process models and expanding the BPMN language [46, 47], enriching business processes with threat information [48], or using such security aware models for specific security purposes like verifying security standard compliance [49]. Playbooks primarily recognize BPMN as a data format for sharing generic playbooks and their visualization [22, 23, 41]. While those also highlight the need for automating playbook execution, they rely on transforming the BPMN playbooks into another format, e.g., CACAO, without discussing the process automation capabilities of BPMN. The potential of BPMN for playbooks is also demonstrated in [50], where a mapping between the playbook CACAO standard and BPMN 2.0 is presented, with the aim of enhancing understandability and comparability in playbook management by bridging different format standardizations. We utilize BPMN as a tool for visualizing advisory-based playbooks, in conjunction with the automated generation approach. Although certain steps of playbooks cannot be automated completely, our method integrates visualization

and step classification to leverage BPMN's strengths in automated playbook generation. While some playbook steps remain manual tasks, our tool-based approach of using BPMN expands related work on using BPMN for playbook visualization.

3.3 Extracting & Structuring Security Processes

The ever-increasing complexity of security data sources requires the extraction of security-relevant information into organizational processes of security management. The extraction of security information from unstructured text using techniques of Natural Language Processing (NLP) is primarily researched in the field of CTI processing and automation, as discussed in [51] and [52] for further details. Transformer-based architectures, e.g., LLMs, got attention of research for processing extensive amounts of data, e.g., vulnerability descriptions in respective databases [53], or even for the extraction of more complex interrelated security concepts [38]. We adapt this research trend for our use case of generating complex security processes based on unstructured security advisories. Research on processing security information is also relevant in the domain of security advisories, where the applied NLP techniques still rely on pattern matching [17, 18]. Our approach contributes to this use case and further enhances it through the orchestration of LLMs, which achieve significantly better performance on the complex data structures inherent in security workflows.

3.4 Vulnerability and Advisory Analysis

Research utilizes novel methods for processing cybersecurity data sources to identify trends and common topics, with the intention of enhancing or rebalancing the defensive efforts of organizations. Vulnerabilities, mostly CVEs, are the most prominent data sources for analyzing threats. In [54] and [55], common topics within vulnerability descriptions were analyzed with the intention of identifying popular attacking trends. In [56], the focus lies on assessing the severity of vulnerabilities by processing the word dependencies of CVE descriptions with a graph convolutional neural network. Other work, e.g., [57] or [58], employed a visual approach based on knowledge graphs for analyzing data grounded on security vulnerabilities, such as CVE or malware databases. Although our dataset is also linked to vulnerabilities, we shift the perspective from attacking themes within vulnerabilities to a defensive point of view within security advisories. A data analysis using a similar dataset is conducted by [59], but here the focus does not lie on the actual contents of the advisories, but the dependencies in between the publishing platforms. In [25], they identify actionable steps within the advisories but conduct their analysis on a significantly smaller dataset. Their work also did not investigate defensive themes in cybersecurity advisories, nor did it analyze the structural organization of the advisories.

4 Playbook Generation System

Figure 1 gives an overview of the proposed playbook generation method. The artifact includes four different components: the *Input*, two LLM orchestration layers, and the *Output*. As an Input, an unstructured security advisory together with the meta-model schema is presented to the first LLM layer. It generates an advisory playbook, which consists of a set of tasks and a control flow. The tasks are then put into the second layer, where they are classified into either a *Service*-Task or a *Human*-Task. Service-Tasks can further be classified into Actions, e.g, in OpenC2-Actions, that can be used for automation purposes. This meta-model is a structured and machine-readable format, e.g., in JSON, which defines the structure of the playbooks, setting different tasks in relation to each other and adding additional information about the playbooks, such as affected products and the vulnerabilities described within the advisory. Based on this meta-model, the LLM generates a playbook, and the BPMN-Parser can visualize it. The combination of a machine-readable meta-model and BPMN visualization constitutes the semi-automated security playbook as an output. A user can now follow the steps within the playbooks, execute Service-Tasks, and visually assess if the playbook is relevant for their specific use case. This improved procedure offers a usability benefit, as the user is not required to analyze the entire advisory for relevant information regarding a vulnerability.

In the following, we provide a detailed description of the central components of the generation method, precisely the meta-model and both of the LLM layers, followed by a prototypical architecture proposition, that implements the introduced concepts. After, that we show an example of how the generated playbooks incorporate the components of the meta-model.

4.1 Meta-model

Since existing playbook formats such as CACAO are overly complex for the purpose of automated playbook generation, we designed a dedicated, task-specific meta-model. This design allows the LLM to concentrate exclusively on the essential information required, without being affected by unnecessary attributes present in broader standards. The *meta-model* is a data structure that defines the necessary components as the foundation behind the playbook generation method. This reflects how the concepts of security advisories in general, e.g., the integration of executable commands, are incorporated with automation and visualization in mind. The format of the meta-model has different requirements: (1)

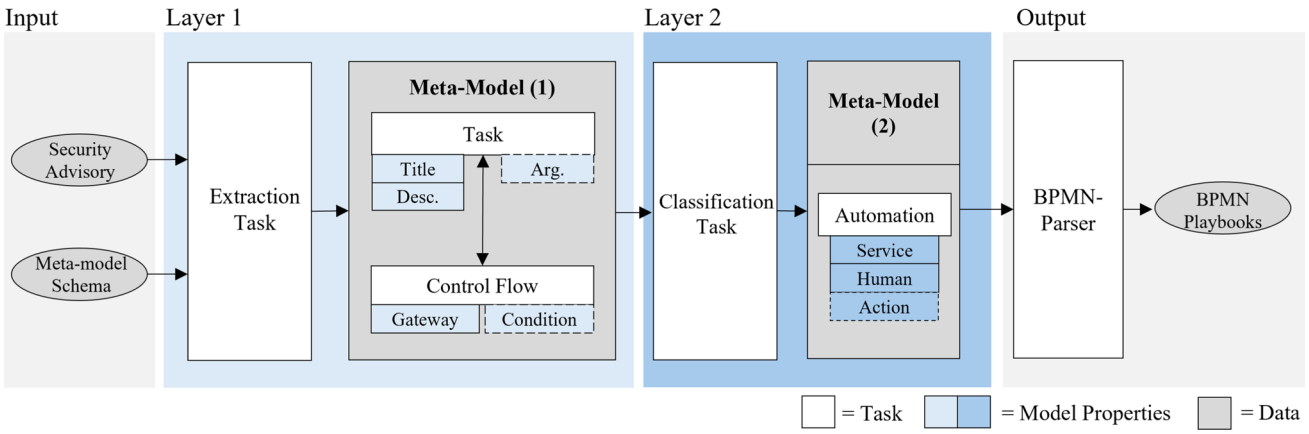

Fig. 1 Overview over the developed artifact, highlighting both of the LLM Layers (1,2) with their Input and Outputs, and the BPMN Parser

Workflow: the data structure must support the implementation of the playbook's workflow, which requires process gateways for conditional branching or parallel steps as well as distinct action components. (2) *Machine readability*: the data format must be machine-readable as it includes automated Service-Tasks that contain commands for certain security systems. Due to these requirements, we chose JSON as the basis for the meta-model. The meta-model consists of the *Steps* and *Additional information*, which can be added in the end. A step can further be divided into an *Action*, which represents tasks of the playbook, and three different gateways. An action consists of: a *step_number*, a *title*, a *description*, optional *commands*, and the *next_step*. A *Gateway* also consists of a *step_number* and the *next_step*, but it differs in *type*, which can be either a *Condition*, *Switch_Condition*, or a *Parallel* gateway. Additional information includes miscellaneous arguments used for categorizing the playbook itself. These mainly consist of a set of *affected_products* and the *CVE_ID* to which the advisory is related. Within the generation method, the meta-model is provided to the first layer and then implemented with a set of *Tasks* (Action steps, Arguments) and *Control Flow* Elements (Condition, Switch-Condition, Parallel). The generated playbook, see. 1 as Meta-Model (1), is then used as the input for the second LLM Layer, where the *tasks* are enriched with the *Automation* classification (Service, Human).

4.2 Layers

The concept of the generation method contains two layers, each addressing a different orchestration task: either extracting the basic playbook and populating the meta-model schema or enriching it by classifying tasks. Dividing the tasks into *Extraction* and *Classification* allows the orchestration framework to provide information specifically to the current LLM task.

Layer 1 - Extraction. As shown in Figure 1, the first layer represents one of the two central components of the concept. It takes the security advisory in the form of unstructured text together with the meta-model schema. The LLM is then orchestrated to analyze the advisory input and fill the meta-model. After the first layer, the meta-model reaches the first state, which consists of a set of *Tasks* and a set of *Control Flow* elements. A task is a combination of a *title*, *description*, and optional *arguments*. The title is a short version of the description, intended for the visualization within the diagram (e.g., BPMN). The control flow consists of a *gateway* type and, in the case of a split gateway, a *condition*, which controls the control flow of the process. An *argument* is an optional part of the task, consisting of executable commands, such as shell scripts, or other forms of actual instructions, such as input for a configuration file. This enables the extraction of executable commands, seamlessly integrating them into playbook steps, where they can be automatically executed during playbook operation.

Layer 2 - Classification. Within the second layer, the *Classification*, the orchestration tasks change. Here, the playbook steps are classified by the LLM regarding automation capabilities, and then the tasks of the meta-model are expanded with the new output. It can be of either type *Service* or *Human*. If a task is classified as a Human-Task, the description of the playbook step suggests that it is primarily a manual task that requires human interaction with each other or the system. The LLM is prompted to identify such tasks if an automation is not viable. e.g., in case of a general guideline for the organization or an awareness plan that involves human interaction. A Service-Task can, in theory, be fully or mainly automated through technical measures. To further accelerate the generated playbook into a playbook with domain-specificity, we also classify tasks into different OpenC2 commands.

4.3 Architecture

Building on this concept, we present our orchestration architecture. We first introduce the employed frameworks and LLM orchestration concepts, and then describe the implementation of our software application. We publish our prototype together with the generated playbook data open-source on Github[2].

LangChain and Pydantic. LangChain [35] is a framework for building applications powered by LLMs, offering compatibility with a wide range of models. This flexibility means our prototype is designed for more powerful models in the future, which can be integrated with minimal modifications. Furthermore, it simplifies the use of LLMs by enabling few-shot prompting and structured output formatting. A custom JSON schema can be configured for a LLM using LangChain, which forces the LLM to generate output in strict accordance with predefined rules. Our schema, called meta-model, is built with elements analogous to BPMN activities and operators, supporting sequential and parallel actions as well as branch-based conditions. LangChain invokes the LLM with an advisory text, and based on our schema, it extracts a logical process workflow.

Pydantic [60] is an open-source Python library for data validation, which we used to implement a comprehensive set of validation checks for the generated playbooks. Among other things, the validations verify that each step has a unique ID, no step references a non-existent subsequent step, the playbook terminates properly, no empty but mandatory attributes exist in steps or gateways, each conditional gateway contains at least two distinct branches, and that all attribute data types are respected (e.g., ensuring that *step_number* is created as an integer rather than a string). By applying these automated validations to each newly generated playbook, we ensure that the LLM conforms to the required syntax of our custom meta-model format. Moreover, the validation framework proved especially valuable during the early stages of development, as it immediately highlighted generation errors, allowing us to refine the system and eliminate issues efficiently in an iterative process.

Few-shot prompting. One of the most effective ways to enhance model performance when generating structured output is by providing the model with examples of the desired output, also known as *few-shot prompting*. To achieve this, we manually constructed three playbooks based on three different cybersecurity advisories, each adhering to our defined schema. We ensured that the advisories and consequently the derived playbooks varied in structure and comprehensively covered all key components of our schema, including actions, conditions, and parallel execution.

[2] https://github.com/security-ratisbona/Sec_Advisory_BPMN_Generator

Architecture Implementation. For the description of our architecture, we adopt the *Software Architecture Description Method* [61], which utilizes UML as a modeling language to depict the structure and operational behavior of a software application. Due to its widespread use, UML is well-suited for representing software architectures in a standardized way.

Figure 2 illustrates the structural aspects of the playbook generation as a UML class diagram. The application is divided into the two main packages *generation* and *model*, and additionally uses an *external* package for third-party libraries. The *model* contains the domain classes. *WorkflowStep* is defined as an abstract class with four concrete subclasses: *ActionStep*, *ConditionStep*, *SwitchConditionStep*, and *ParallelStep*. The class *Playbook* has a composition relationship with *WorkflowStep*, where the list instance variable *workflow* contains one or more logically connected steps. The lifecycle of each *WorkflowStep* instance is bound to its corresponding *Playbook*, meaning that all steps are deleted when the playbook is removed. Both *Playbook* and *WorkflowStep* inherit from *BaseModel* provided by the Pydantic library, enabling their use as schemas for automated validation. The model classes possess *check()* methods annotated with Pydantic validator decorators to perform syntactic validation of generated playbooks. These include both high-level validations at the playbook level and specialized checks for individual workflow step types.

The *generation* package consists of two classes. The *LLMClient* provides functionality for interacting with LLMs, allowing the use of different LLMs and tasks. *PlaybookGenerator* orchestrates the generation process using the *LLMClient* by reading CERT advisory files, triggering their transformation into playbooks, and storing the generated output. It further offers functionality to invoke the classification of workflow steps after a playbook has been successfully generated. The *external* package includes the most relevant third-party libraries. For clarity, commonly used standard Python libraries such as *typing* and *csv* are omitted from the diagram.

Figure 3 illustrates the operational interactions between the involved objects during the playbook generation process. First, the *PlaybookGenerator* reads the CERT advisory files. It then configures the *LLMClient* by providing the *Playbook* class as a schema for the output *(setup_structured_output_schema(Playbook))*. In addition, it defines a prompt template consisting of a general system prompt and a set of few-shot examples. This template is reused for all subsequent LLM invocations. The *PlaybookGenerator* then iterates over all loaded advisories and triggers the generation process by calling *invoke_LLM(advisory_text)*. For each call, the advisory text is appended to the predefined prompt template, and the resulting prompt, together with the associated *Playbook* schema, is sent to the LLM API. The LLM generates a playbook and returns it in JSON format. The returned output

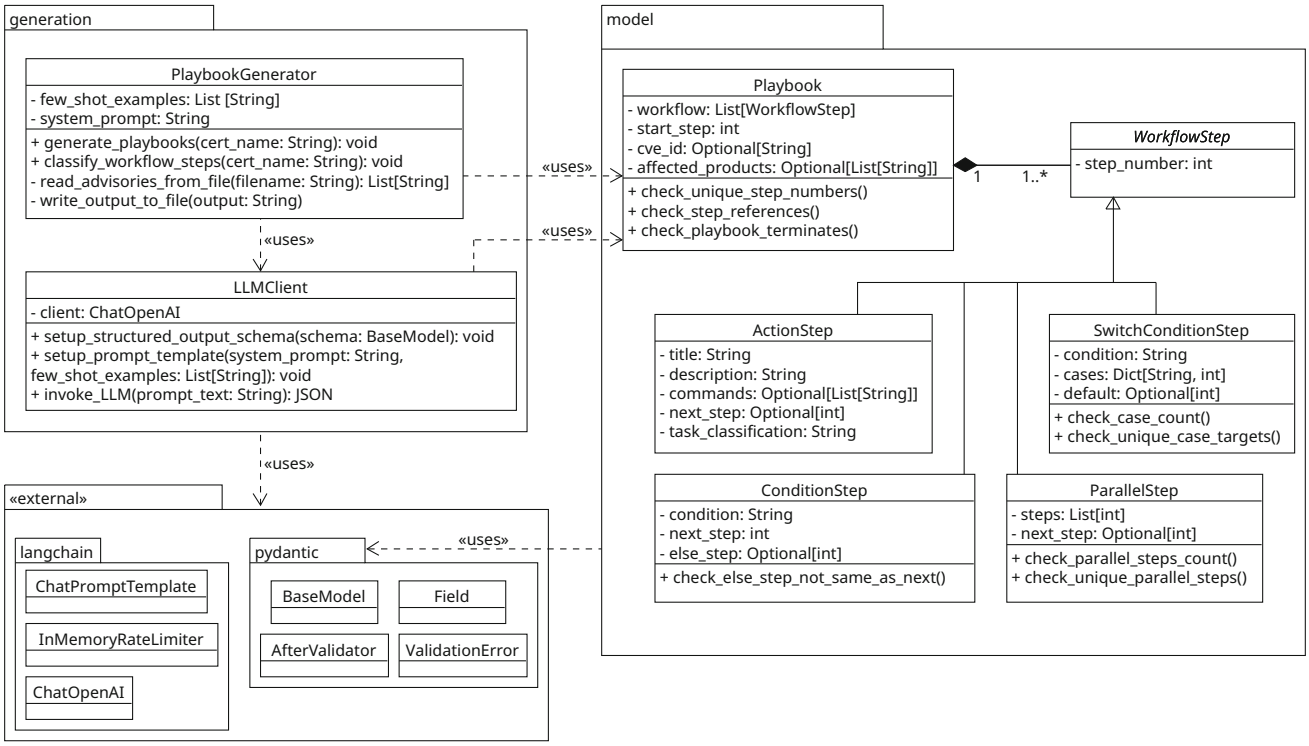

Fig. 2 UML class diagram describes the classes and their relationships for the playbook generation

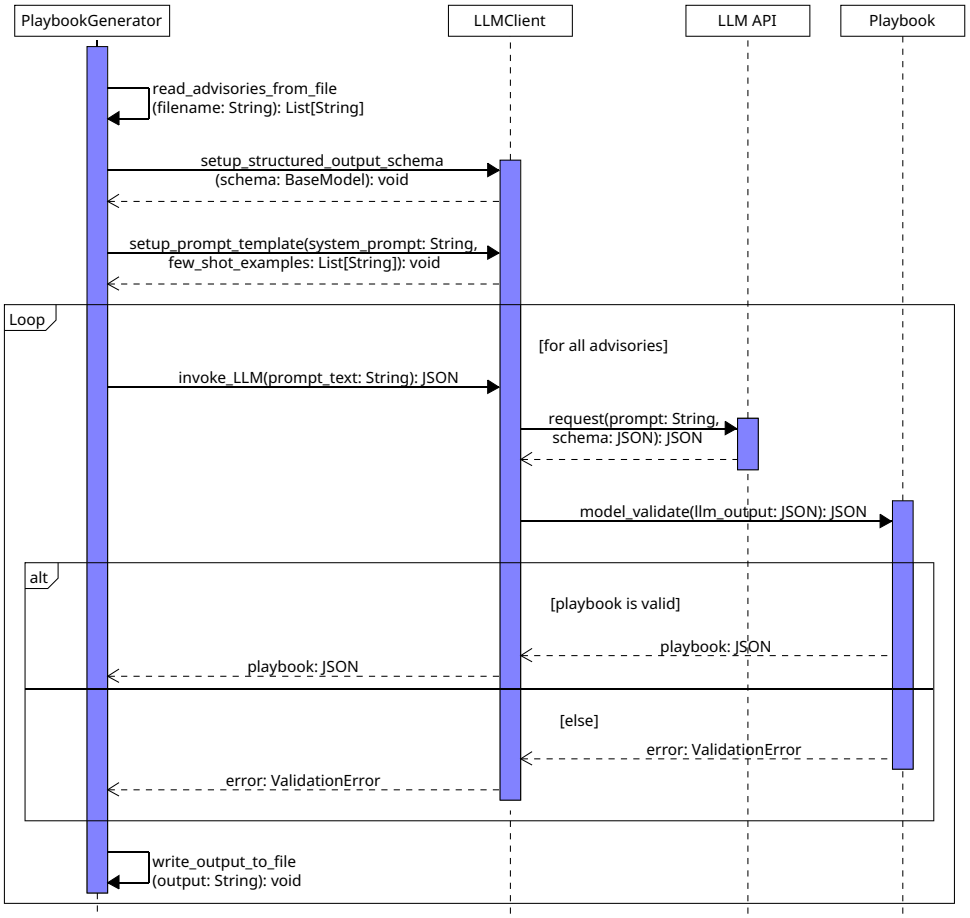

Fig. 3 UML sequence diagram shows the object interactions over time for the playbook generation

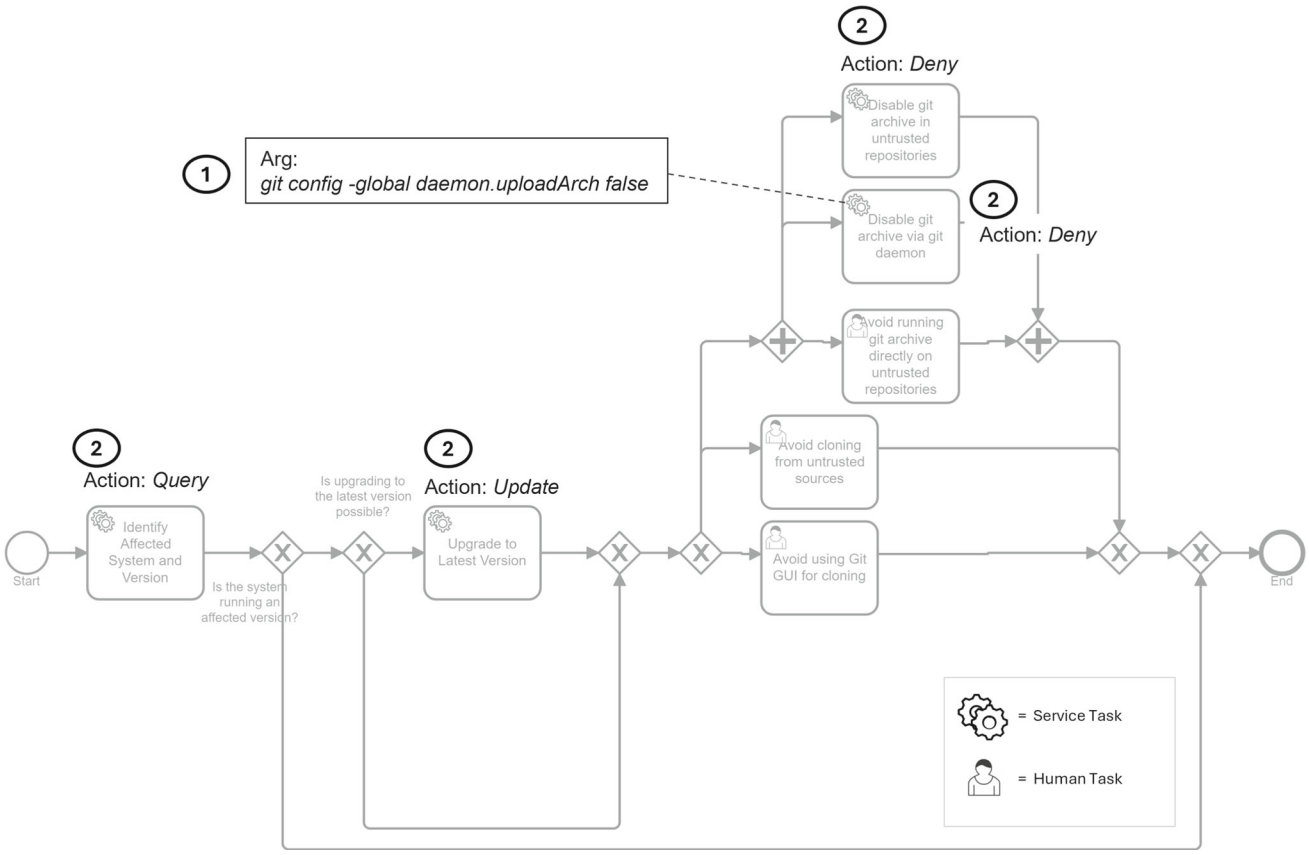

Fig. 4 An example playbook generated by the system, with visual annotations, that are not part of the BPMN visualization, but can be found within the meta-model of the playbook. (1) An *Argument (Arg.)* as executable command. (2) The classification of Human/Service-Tasks and OpenC2 actions

is immediately validated against the schema. This involves executing the *check()* methods defined in the model classes, both at the *Playbook* level and for all contained *WorkflowStep* instances, which are linked via the composition relationship (see class diagram in Figure 2). For clarity, the individual validation method calls are omitted from the sequence diagram. If the generated playbook satisfies the schema constraints, it is returned as a valid result. If any violation is detected during validation, a *ValidationError* exception is raised and returned instead. In both cases the *PlaybookGenerator* persists the output to a file.

4.4 Example: Playbook Use Case

To demonstrate how the meta-model elements are represented within the generated playbooks, we chose an example playbook out of the generated dataset, which includes all possible components defined within the meta-model (see Fig. 1). In Figure 4, we present the example as processed by Layer 1, Layer 2, and the BPMN parser, with additional annotations added for clarity. In Layer 1, the prototype extracts the *Tasks* and *Gateways* from the advisory and sets them into relation, which forms the *Workflow* of the playbook. If there are specific commands or instructions within the advisory, they are extracted as *Arguments (Arg)*. If Layer 2 recognizes a step as a Service-Task, such arguments can be included in the meta-model to form an executable command. The JSON meta-model of the playbooks is enriched with the OpenC2 *Actions*. Figure 4 also shows the classified *Actions* for each of the Service-Tasks, within the context of the playbooks use case. At last, the meta-model is used as the basis for visualizing the playbook via the *BPMN-Parser*, resulting in the displayed example.

5 Evaluation

We show that orchestrating LLMs for the task of playbook generation allows organizations to integrate unstructured advisory data into their security risk management processes. This contribution is evaluated with an experiment study [62], where we implement our orchestration architecture and process a representative dataset of advisories. Additionally, we evaluate the performance of the LLM classification task,

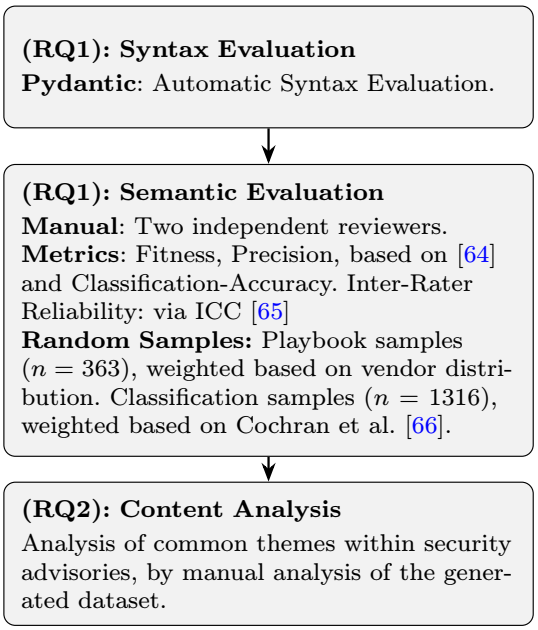

Fig. 5 Overview of the two-fold evaluation approach and subsequent content analysis

which represents a step towards automating security playbooks. As a result of our evaluation efforts, we publish the resulting dataset of semi-automated playbooks together with the full prototypical implementation.

Figure 5 gives an overview of the structured approach behind the evaluation and the subsequent content analysis. At first, we check the generated playbooks with Pydantic on the performance of the LLM to fit the correct meta-model schema, therefore performing a syntax check. The second evaluation step consists of a manual qualitative evaluation of the semantics of the generated set of playbook process models. For this step, we draft a random sample weighted to the advisory vendors and calculate specific metrics that provide an estimate of how well the generated playbook represents the advisory. Those metrics provide an answer to how well we can extract the playbooks regarding RQ1. Finally, we analyze the contents of the generated playbooks to address RQ2, focusing on the identification of themes and security processes that are characteristic of advisory-based playbooks. In the following, we describe our experiment study in detail and provide limitations of our architecture and evaluation.

5.1 Experiment Setup

Data Collection. We collected a set of security advisories from different CERTs that provide complex advisories to conduct the experiment study. Security advisories are published in different formats that vary in the level of structuring. Formats like CSAF are becoming increasingly prominent, but to date, they lack broad adoption within CERTs. Therefore, we only included unstructured advisories for the evaluation of the method's performance. The advisories were published on the websites of the different CERTs, and crawled by collecting the HTML content for extracting advisory metadata and descriptions. The resulting dataset of advisories consists of 759 advisory descriptions from three different vendors (CERT-EU [19], VDE-CERT [29], CISA [63]). We chose this set of CERTs because they are well-known, provide their advisories in the English language, in unstructured formats (HTML or plain text), and also have a reasonable number of recent advisories. Table 1 shows the distribution of the advisories over the different vendors making up the dataset of the experiment.

Table 1 The dataset distribution over the different security advisory vendors (Total: n=759)

Source	Formats	Amount	Years
EU-CERT	HTML	253	22-25
VDE-CERT	HTML	102	17-25
CISA SA	CSAF, JSON, HTML	134	23-25
CISA ICS	CSAF, JSON, HTML	270	23-25

Data Preparation. By scanning through the contents of the collected dataset, we identified relevant sections for our playbooks. The most important parts are the sections describing the *Mitigation* of the presented vulnerabilities. In addition, some other types of information were valuable for the playbooks. Those included, e.g. *Best Practices*, *Monitoring Techniques*, *Workaround*, and *System Hardening*. Based on those headings, we extracted the content from the HTML documents and created a set of CSV files linking the contents to the respective advisories.

Experiment. Our experiment was orchestrated by a Python-based prototype implementing the architecture. For the foundation model of the LLM, we relied on the OpenAI GPT-4o model, which we prompted via the API. We chose the model as it was affordable and provided good performance metrics for NLP tasks during the time of this work. Overall, the generation of the playbooks created computing costs of 17.44$. The collected dataset of advisories was provided to the GPT model and supported by the orchestration architecture. LangChain was used to provide our meta-model schema to the LLM. At first, we tried generating a small set of playbooks (approx. 20) to adjust the quality and quantity of the few-shot prompts, the meta-model, and the prompts accordingly until the architecture provided stagnant results. We found that our architecture achieved the best balance between resource cost and performance when using three diversely structured few-shot prompts. Then we generated the full dataset of playbooks in our JSON meta-model. In addition, we classified the full set of 4563 playbook steps from among the playbook dataset into Human and Service-Tasks.

To demonstrate the classification of OpenC2, a random subset of Service-Tasks (400 tasks) was then further classified into the possible action classes. The whole set of generated playbooks was then visualized by a self-developed parser, which takes the JSON meta-playbook as an input and creates the BPMN. Both sets of playbooks, in the JSON- and in the BPMN representation, and the code of the prototype can be found within our GitHub[3].

5.2 Analysis

General Approach. The experiment results were then analyzed as depicted in Figure 5. First, the resulting dataset was checked by Pydantic to ensure that the *syntax* matches the schema of the meta-model. Second, the workflow and the classifications were evaluated for *semantic performance*. For the workflow, we drafted a random, weighted sample of 363 advisory-to-playbook pairs, ensuring that the sampling reflected the proportional distribution of playbooks among the various CERT vendors. For each advisory, a manually crafted playbook was extracted by analyzing the relevant text sections and then comparing it with the playbook generated by our architecture. Two authors independently compared the generated playbooks with the advisories and extracted the indicators required for the qualitative metric calculation. The resulting datasets, containing the Fitness and Precision values for each evaluated playbook, were then used to compute the Intraclass Correlation Coefficient (ICC), which provides a measure of inter-rater reliability between the two independent semantic evaluations [65]. While the single-rater metric (ICC(2,1)) quantifies the reliability of an individual reviewer's rating, the average-rater metric (ICC(2,k)) quantifies the reliability of the mean rating across all k reviewers.

For evaluating the classification accuracy of the Human and Service classified steps, we drafted a statistically representative random sample ($n = 1316$) for our dataset based on Cochran's formula for categorical data [66] (99% confidence level with 3% margin of error). Finally, the set of evaluated playbooks was manually analyzed for common defensive themes.

Semantic Evaluation Approach. Evaluating the quality of generated process models is commonly addressed in process mining through conformance checking, which compares real-world event log data with a process model to measure how well the model reflects observed behavior [64, 67]. However, the evaluation setting in this paper differs fundamentally from traditional conformance checking. Security advisories are unstructured textual descriptions rather than structured event logs, and the generated playbooks represent interpreted workflows derived from natural language. As a result, the automated alignment of event logs with process models required for formal conformance checking is not feasible in this context without extensive preprocessing and additional assumptions about event extraction. Instead of applying process mining techniques directly, we only adopt the general idea of comparing a generated process model with its source representation to evaluate how well the semantics of the advisory are preserved in the generated playbook. Our evaluation therefore follows a human-centered approach, in which we manually compare generated playbooks with the corresponding advisories. To structure this comparison and ensure consistency across evaluators, we designed a set of indicators inspired by the conceptual notions of *fitness* and *precision* commonly used in automated conformance checking. We adopt the terminology of fitness and precision, as these concepts are well-established in the evaluation of process models and provide an intuitive framework for assessing completeness and accuracy. In this paper, we interpret these concepts at a semantic level. Fitness captures how completely a generated process model represents the actions described in the advisory, while precision reflects how accurately the model avoids introducing unsupported or unnecessary elements. Table 2 shows the self-designed indicators and their calculation within our evaluation: The number of correctly represented *Steps*, the number of correct *Gateways*, the number of *Summarized* steps, and the number of *Extended* steps. *Summarized* steps describe cases where the LLM combines two distinct advisory actions into a single task. *Extended* steps refer to tasks introduced by the model that are not supported by the advisory text. These indicators allow different evaluators to systematically assess the semantic correspondence between advisories and generated playbooks and derive quantitative values for the fitness and precision metrics.

5.3 Limitations

Our work faces some limitations, which we will present here. First, the research time does not match the rapid development in the field of LLMs. As a result, we implemented a thorough evaluation approach, a meta-model of advisory-based playbooks, and a conceptual description of the method, which enhances reproducibility with other comparable LLMs. In a similar regard, resource limitations caused by API costs constrain the experiment's scope. Because of data pre-processing and the orchestration architecture, we are still able to construct a statistically relevant dataset of playbooks. Second, the manual evaluation approach might lead to a subjectivity bias. We mitigate this bias by using two independent reviewers and by calculating inter-rater reliability. Third, the targeted choice of security advisories as the underlying data source for playbooks reflects a limited view of the topics relevant to playbooks for organizations. Advisories in their current form provide a valuable source for automating security risk

[3] https://github.com/security-ratisbona/Sec_Advisory_BPMN_Generator

Table 2 Overview over the qualitative metrics used for the semantic evaluation of the experiment. Indicators: (Sum) Summarized Steps in Model; (Ext) Extended Behavior in Model; (*) Metric adapted for this Use Case based on [64].

Metric	Description	Formula
Fitness*	The metric describes if the observed behavior within the advisory, is present in the generated process model (Playbook). If the amount of correct steps and gateways in the generated playbook equals the total amount of steps and gateways within the advisory, it recreates the complete behavior of the advisory.	$\text{Fitness} = \frac{\text{No. correct steps and GWs (Model)}}{\text{No. of steps and GWs (Advisory)}}$
Precision*	The metric describes how precise the process model (Playbook) matches the advisory. Irrelevant steps (Ext.) and summarized steps (Sum.) would lead to a lower precision score.	$\text{Precision} = \frac{\text{No. of correct steps (Model)}}{\text{No. of correct steps (Model)} + \text{Sum.} + \text{Ext.}}$

assessment and response to threats, as they focus on mitigation and remediation processes. Future research could expand our work by identifying additional data sources like vendor bulletins, or structured advisory formats like CSAF, for playbook generation. Finally, since the OpenC2 language is still under development, the presented classification of OpenC2 actions is used solely for conceptual advisory analysis rather than for generating generic executable OpenC2 commands. Instead, we extract executable commands directly from the advisories and incorporate them into the generated playbooks, where they can be executed automatically if approved by the analyst. While these commands may be specific to a given system or environment, this limitation is mitigated by the playbooks' ability to differentiate between alternative commands through process gateways, selecting the appropriate one for the respective environment whenever the advisory provides multiple options.

6 Results

In the following, we present the experimental results estimating the architecture's performance in representing advisory tasks within the generated playbooks, as well as the content analysis of defensive strategies and security themes in security advisories.

6.1 Syntax Performance

The evaluation demonstrates that our architecture is capable of generating meta-model–compliant outputs for 725 of the 759 playbooks (95.5%). This approach, based on Pydantic, ensures that the LLM limits the generated playbooks in format to the model. For automation reasons in further processing, it is necessary that the playbooks match the meta-model.

6.2 Semantic Performance

Out of the 725 generated playbooks that passed the syntax compliance check, the random sample of 363 playbooks was selected for a subsequent evaluation of their semantic correctness.

Fitness. Both reviewer A and B achieved similar average scores in Fitness (F) ($F_A = 0.9100$ & $F_B = 0.9154$). Inter-rater reliability, assessed with ICC [65], showed good single rater reliability (ICC(2,1) = 0.74, CI 95% [0.65-0.82]), in combination with an excellent average rater reliability (ICC(2,k) = 0.85, CI 95% [0.79–0.90]). These average ICC results indicate a robust agreement in between the two reviewers. The evaluation of the Fitness (see Table 2) shows that the architecture achieves an overall score of 0.9127, indicating that 91.27% of steps and gateways that should be present within the playbooks are indeed present. This score varies depending on the different ways the CERTs build their advisories. While *EUCERT* (92.92%) and *CISA* (90.00%) combine update recommendations with general best practice advice, *VDE CERT* includes more concrete steps to mitigate or remediate vulnerabilities. Such concrete commands have to be executed in isolation in some cases, whereas the LLM extracted them as part of other steps. Due to our evaluation approach we had to flag such steps as false, resulting in a lower score of 88.39%.

Precision. Agreement in between reviewer A and B achieved sufficient inter-rater reliability when calculating Precision (P) (Single rater: ICC(2,1) = 0.5359 & CI 95% [0.39, 0.66].

Avg. rater: ICC(2,k) = 0.70, CI 95% [0.56–0.79]). Single rater reliability indicates that the single reviewer's evaluation may show some subjectivity. However, the average rater reliability shows, that aggregating multiple reviewers helps balancing the subjectivity to achieve more robust results. This can be explained with the Precision metric requiring more interpretation, as judging certain steps as summarized or extended is more influenced by the context of the advisory. The evaluation of the Precision shows that the correctly classified steps match the descriptions of the advisory, achieving a value of 0.9533 ($P_A = 0.9608$ & $P_B = 0.9458$). As this value is close to the optimal Precision of 1, it can be said that the LLM accurately reflects the strategies from the advisories. In detail, the deviation from the perfect score is primarily influenced by the set of extended steps present only in the generated playbook. The indicator of summarized steps within the playbooks was only present in a few cases, leading to a low impact on the average precision score.

Classification. The evaluation of the Classification results shows that the model is able to correctly classify a playbook step in the two classes *Human* and *Service* in 83.28% of cases. Error analysis shows that some tasks are not specifically fitting for one of the categories. Service-Tasks make up approximately 63% of the 4563 tasks, which supports our observation that advisory-based playbooks can be automated to a significant extent.

6.3 Content Results

Security Advisories. By transforming the advisories into security playbooks, we were able to examine the evaluation sample of playbooks within the generated dataset and present key indicators and content topics present in advisories. Figure 6 presents the most common security themes within security advisories. Because of the vulnerability-centered nature of advisories, *Update* is the most prominent category of topics with around 65.4% of simple update playbooks and an additional 8.53% of complex update playbooks. Especially the complex update playbooks benefit from the automation within our meta-model, as version-dividing gateways can be used with the additional information within the playbook, to query the correct update for the domain system. Figure 6 also highlights topics among playbook steps that present more general advice than concrete remediation or mitigation techniques. So do 38.86% of playbooks include *General Guidelines*, from either the system vendors or the CERTs. We found that the provided general guidelines offer broad recommendations that can assist organizations in enhancing their overall security posture. Those are often provided in the end, but also mixed into the vulnerability mitigation strategies. Such guidelines could, for example, include the proposition of a defense-in-depth approach or a general risk assessment strategy. Under this category, there are also topics

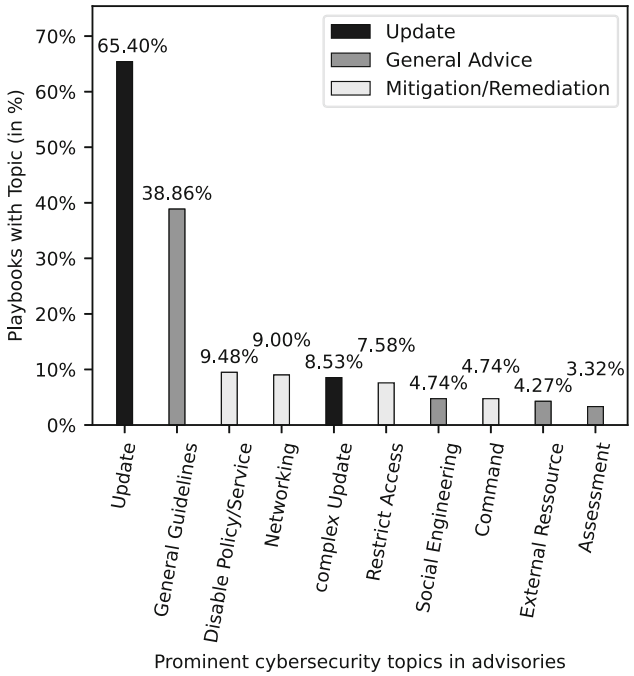

Fig. 6 Overview of the top 10 security themes identified in cybersecurity advisories, as reflected in our generated playbooks and grouped into three overarching categories

that are related to raising awareness about *Social Engineering* (4.74%) or referring to *External Resources* (4.27%) for additional vendor-specific advice. Other topics within playbooks fall into a category that specifically includes security measures that provide a complete *Remediation* of a vulnerability or *mitigate* the risk further. The most prominent topics within playbooks include the *disabling* of security-related policies within software systems or completely disabling a vulnerable service (9.48%). Concrete *networking* measures, like e.g., blocking certain IPs or isolating vulnerable systems, are present in 9% of playbooks. The topic of restricting access of users to a vulnerable system is present in 7.58% of playbooks. Based on the analysis, only 4.74% of playbooks contain specific executable commands or configuration file modifications to automate advisory implementation.

Actionable Steps. The consequent classification of the Service-Tasks into the action categories of the OpenC2 specification, allows the generated playbook to take a step further to the automation of the playbook execution. Automating playbooks requires two parts: (1) Enriching the basic playbook with domain context and (2) Defining ActionTarget pairs, e.g. described within CACAO [25, 33], thus allowing executable commands (Action) to be carried out on the respective target systems (Target). Our classification into OpenC2 actions, shows similar action themes compared to our analysis of advisory contents. Each of the classified Service-Tasks was assigned one, two, or three different actions. This was necessary, as some tasks are written on

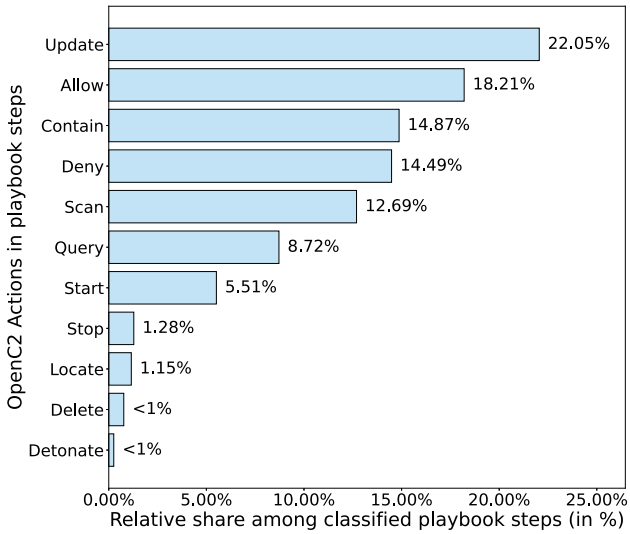

Fig. 7 Relative Distribution of classified Service-Tasks among OpenC2-Action-Classes

a high level that matches multiple action classes. Figure 7 shows the distribution over the most common OpenC2 Action classes within the demonstrating set of Service-Tasks. Fittingly, the most common action is *Update*, which makes up 22.05 % of tasks. As more than 70% of playbooks include updates as security theme, and only 22% of steps are classified as *Update*, shows, that playbooks based on advisories often include Update, but in addition evolve around additional tasks. *Deny*, preventing an event or process flow, and *Contain*, isolating a file, process, or entity, and *Stop*, halting a system or activity, together make up almost a third of the Service-Tasks. They represent the mitigation and remediation strategies in case an update is not a viable option for a certain vulnerability or additional defensive measures are useful. Such actions are often concluded by an *Allow* task, which includes permitting access or the execution of a targeted system. This shows, that the generated playbooks benefit from the conditional gateway logic within the workflow, as preventing system actions is often followed by restoring the original access and execution rights, after the vulnerability is remediated or at least mitigated. Additional actions found within the dataset include *Query*, initiating information request, or *Scan* activities. These are often included within the playbooks in form of tasks that either execute vulnerability scans of a targeted system or by querying external resource bases, e.g., malware databases or vendor specific mitigation strategies.

7 Discussion

In regard to RQ1, the results of the experiment show that LLMs can provide good results in extracting workflows out of unstructured text and structuring them within a meta-model. Based on the generated dataset, we were able to analyze the contents of the security advisories to discuss advisories as a source for structured semi-automated security playbooks. Based on our analysis of the dataset and the subsequent content analysis, we present recommendations and best practices for CERTs to improve the security advisory publishing, as well as for organizations in processing them in form of advisory-based playbooks.

Advisory based playbooks link automated updating tasks with additional mitigation strategies.

Security advisories are often based on vulnerability data, which requires the playbooks to check either if the CVE-ID is relevant for the system or to check specific affected products. The results show that our playbook dataset includes such a check at the beginning of the process model in 65.40% of cases. Advisories often explicitly mention the relevant CVE or products, or just provide them in the title of the advisory. Playbooks derived from cybersecurity advisories encompass a wide range of defensive measures, such as system updates, configuration changes, attack detection, and temporary workarounds, designed to address newly discovered vulnerabilities or situations where patching is not immediately feasible (e.g., in industrial environments or healthcare settings). These playbooks can be categorized into several types: simple update playbooks, update playbooks that include additional mitigation steps, playbooks combining update steps with temporary workarounds when patching is not possible, and playbooks that provide only mitigation strategies.

A playbook outlines the entire response process against vulnerabilities or threat actor groups in detail, while also incorporating alternative paths of action. This process flexibility allows for granular control and the customization of countermeasures based on an organization's specific context, for instance, determining whether a patch can be applied immediately or if a temporary workaround must be implemented first. Simple update playbooks frequently involve a high number of Service-Tasks, many of which can be automated using the *Update* action of OpenC2 in combination with the affected products. Moreover, a well-structured playbook integrates seamlessly into an organization's existing

systems, coordinating with them to effectively remediate vulnerabilities. While Vulnerability Management Systems detect and prioritize vulnerabilities across the infrastructure, the playbook supports experts by finding appropriate mitigation strategies and orchestrating their implementation via a BPMN process engine such as Camunda. In this way, the playbook can interact with other security systems, e.g., triggering firewall configuration changes or instructing a patch management system to deploy updates across the organization.

CERTs mix vulnerability mitigation and remediation with general guidelines.

The benefit of structuring and visualizing complex advisories lies in efficiently finding the required update path, or, if not possible, fitting mitigation or remediation strategies. Our playbook data show that CERTs tend to mix valuable information, such as changing vulnerable security policies, with general security guidelines in 38.86% of the generated playbooks. This obfuscation of relevant tasks makes extracting the relevant and time-critical information from the advisories more time-demanding. Such general guidelines should be published separately, rather than being mixed into the mitigation, remediation, or solution sections of advisories. Especially in the ICS field, advisories show more general guidelines than specific security measure improvements, which hinders security experts in finding effective and time-critical threat solutions. General guidelines for existing security management systems may be beneficial for enhancing overall security. However, in the context of mitigating a vulnerability, they can be seen as a barrier to automated threat response.

Automating security tasks still requires human interactions, as central process steps require contextual assessment and authority.

Our analysis reveals that 63% of the generated playbook steps are Service-Tasks and could thus be automated by leveraging the corresponding OpenC2 action categories. Currently, such tasks remain dependent on human interaction, as they require enrichment with the domain context of an organization's systems. Furthermore, only 4.74% of playbooks contain executable commands, scripts, or present instructions for changing system properties. Organizations like CERTs should focus more on providing specific instructions that address concrete threats, rather than offering general service-oriented advice that still requires human interaction. With the approach presented in this paper, organizations can take a step to automate Service-Tasks, e.g., disabling a specific security measure, and enrich them with manual tasks, e.g., talking to domain experts. The combination of actionable tasks and human control instances constitute as a semi-automated approach in the form of advisory-based playbooks. In the future, human operators could supervise the process by monitoring playbook execution, evaluating conditional paths and mitigation strategies, and authorizing their implementation as needed.

8 Conclusion

In this paper, we presented an LLM orchestration framework for transforming unstructured security advisories into BPMN-visualized security playbooks, enabling the structured representation of defensive procedures extracted from textual advisories. Furthermore, we classified the generated playbook steps based on their automation potential, distinguishing between steps that can potentially be executed automatically and those that require human involvement. We implemented and evaluated our architecture using a dataset of 725 security advisories. The evaluation included assessing the quality of the generated playbooks, their automation potential, and analyzing security-related topics contained in the advisories. Using a human-centered evaluation approach, we assessed the alignment between generated playbooks and their corresponding advisories. The results demonstrate that LLM-based playbook generation can produce high-quality results in this context. The analysis of the automation potential for executing the generated playbooks revealed that more than half of the generated playbook steps can be classified as potentially automatable. However, several steps in the mitigation process remain inherently manual and require human assessment, thereby preventing the full automation of playbook-based vulnerability mitigation. Our analysis further shows that advisories primarily focus on vulnerability updates supplemented by general security guidelines, rather than detailed mitigation strategies. Future work could build on our methods in different ways, by analyzing a bigger set of advisories, including other data formats, e.g., threat reports, or by using state-of-the-art LLM technology. Furthermore, combining our meta-model-based approach with playbook standardization formats, such as CACAO, could benefit the automation component of the playbooks. In addition, the generated playbooks should be investigated for practicality, meaning how much additional effort it would take to enrich them with domain specific context for organizational deployment and how much they would improve organizational vulnerability mitigation processes.

Author Contributions All authors contributed to the concept and design of this work. D.O. and J.G. contributed to the data collection, analysis, implementation, and the writing of the main manuscript. All authors read and approved the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. Partial financial support provided by the DEFENSIVE project and the REINSIST project. Funded by the German Federal Ministry of Research, Technology, and Space (16KIS1568K) and the Bavarian Ministry of Economic Affairs, Regional Development and Energy (DIK0741/02).

Data Availability The generated data, together with the source-code of the prototypical architecture are published in our GitHub. It can be accessed with this url: https://github.com/security-ratisbona/Sec_Advisory_BPMN_Generator

Declarations

Competing interests The authors declare no competing interests.

References

1. Bo, L., Meng, X., Sun, X., Xia, J., Wu, X.: A comprehensive analysis of nvd concurrency vulnerabilities. In *2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS)*, 9–18, (2022)
2. Kumar, A., Fung, C.: An analysis on cve vulnerabilities of the internet of things. In *2024 7th Conference on Cloud and Internet of Things (CIoT)*, pages 1–6, (2024)
3. International Organization for Standardization. ISO/IEC 27001: Information Security, Cybersecurity and Privacy Protection – Information Security Management Systems – Requirements. https://www.iso.org/isoiec-27001-information-security.html, 2022. Accessed: 2025-02-18
4. Joint Task Force Transformation Initiative. Managing Information Security Risk: Organization, Mission, and Information System View. SP 800-39, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, March (2011)
5. Serror, M., Hack, S., Henze, M., Schuba, M., Wehrle, K.: Challenges and Opportunities in Securing the Industrial Internet of Things. IEEE Trans. Industr. Inf. **17**(5), 2985–2996 (2021)
6. Anton, S.D.D., Fraunholz, D., Krohmer, D., Reti, D., Schneider, D., Schotten, H.D.: The global state of security in industrial control systems: An empirical analysis of vulnerabilities around the world. IEEE Internet Things J. **8**(24), 17525–17540 (2021)
7. Adaros Boye, C., Kearney, P., Josephs, M.: Cyber-risks in the industrial internet of things (IIoT): towards a method for continuous assessment. In *Information Security: 21st International Conference, ISC 2018, Guildford, UK, September 9–12, 2018, Proceedings 21*, pages 502–519. Springer, (2018)
8. Kure, H.I., Islam, S., Mouratidis, H.: An integrated cyber security risk management framework and risk predication for the critical infrastructure protection. Neural Comput. Appl. **34**(18), 15241–15271 (2022)
9. Reittinger, T., Grill, J., Pernul, G.: Share and benefit: incentives for cyber threat intelligence sharing. Int. J. Inf. Secur. **25**(1), 37 (2026)
10. Miranda, L., Vieira, D., Nogueira, M., Ventura, L., Bicudo, M., Martins, M., Senos, L., de Aguiar, L.P., Lovat, E., Menasche, D.: Measuring and modeling software vulnerability security advisory platforms. In *Risks and Security of Internet and Systems: 15th International Conference, CRiSIS 2020, Paris, France, November 4–6, 2020, Revised Selected Papers 15*, pages 31–48. Springer, (2021)
11. Shafqat, N., Masood, A.: Comparative analysis of various national cyber security strategies. International Journal of Computer Science and Information Security **14**(1), 129–136 (2016)
12. West-Brown, M.J., Stikvoort, D., Kossakowski, K-P., Killcrece, G., Ruefle, R., Zajicek, M.: *Handbook for computer security incident response teams (CSIRTs)*. Carnegie Mellon University, Software Engineering Institute, (1998)
13. Schlette, D., Caselli, M., Pernul, G.: A comparative study on cyber threat intelligence: The security incident response perspective. IEEE Communications Surveys & Tutorials **23**(4), 2525–2556 (2021)
14. Riebe, T., Kaufhold, M.-A., Reuter, C.: The impact of organizational structure and technology use on collaborative practices in computer emergency response teams: An empirical study. Proceedings of the ACM on human-computer interaction **5**(CSCW2), 1–30 (2021)
15. Kaufhold, M-A.: Thea Riebe, Markus Bayer, and Christian Reuter. 'We Do Not Have the Capacity to Monitor All Media': A design case study on cyber situational awareness in computer emergency response teams. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–16, (2024)
16. OASIS Common Security Advisory Framework Technical Committee. Common Security Advisory Framework (CSAF) Version 2.0. Technical report, OASIS, November 2022. OASIS Standard
17. Kaufhold, M-A., Bäumler, J., Koukal, N., Reuter, C.: Towards a security advisory content retrieval and extraction system for computer emergency response teams. Mensch und Computer 2024 - Workshopband, (2024)
18. Ramnani, R.R., Shivaram, K., Sengupta, Annervaz K. M.: Semi-automated information extraction from unstructured threat advisories. In *Proceedings of the 10th Innovations in Software Engineering Conference*, ISEC '17, page 181–187, New York, NY, USA, (2017). Association for Computing Machinery
19. CERT-EU. Security advisories, 2025. Accessed: 2025-02-20
20. Baumer, T., Grill, J., Adan, J., Pernul, G.: A trust and reputation system for examining compliance with access control. In *Proceedings of the 19th International Conference on Availability, Reliability and Security*. Association for Computing Machinery, (2024)
21. European Parliament and Council of the European Union. Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS2 Directive), December (2022). Accessed: 2025-02-20
22. Akbari Gurabi, M., Nitz, L., Bregar, A., Popanda, J., Siemers, C., Matzutt, R., Mandal, A.: Requirements for playbook-assisted cyber incident response, reporting and automation. *Digital Threats*, 5(3), (2024)
23. Schlette, D., Empl, P., Caselli, M., Schreck, T., Pernul, G.: Do you play it by the books? a study on incident response playbooks and influencing factors. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 3625–3643. IEEE, (2024)
24. Romano, J., Sparrell, D.: Open Command and Control (OpenC2) Language Specification Version 1.0. Committee Specification 02, OASIS Open Command and Control (OpenC2) Technical Com-

mittee, 2019. Latest version available at https://docs.oasis-open.org/openc2/oc2ls/v1.0/oc2ls-v1.0.html
25. Empl, P., Schlette, D., Stöger, L., Pernul, G.: Generating ics vulnerability playbooks with open standards. Int. J. Inf. Secur. **23**(2), 1215–1230 (2024)
26. National Institute of Standards and Technology (NIST). National Vulnerability Database (NVD). https://nvlpubs.nist.gov/nistpubs/Legacy/SP/, 2025. Accessed: 2025-02-18
27. European Union Agency for Cybersecurity (ENISA). European Union Vulnerability Database (EUVD). https://euvd.enisa.europa.eu/, 2025. Accessed: 2025-07-16
28. Wunder, J., Aurich, J., Benenson, Z.: From Chaos to Consistency: The Role of CSAF in Streamlining Security Advisories. In *Proceedings of the 2024 European Symposium on Usable Security*, pages 187–199, (2024)
29. CERT@VDE. Security advisories, 2025. Accessed: 2025-02-28
30. Schneider Electric. Security advisories, 2025. Accessed: 2025-02-28
31. Empl, P., Schlette, D., Zupfer, D., Pernul, G.: SOAR4IoT: securing IoT assets with digital twins. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–10, (2022)
32. Gurabi, M.A., Nitz, L., Castravet, R-M., Matzutt, R., Mandal, A., Decker, S.: From legacy to standard: Llm-assisted transformation of cybersecurity playbooks into cacao format. arXiv preprint arXiv:2508.03342, (2025)
33. Jordan, B., Thomson, A.: CACAO Security Playbooks Version 2.0. Committee specification, Organization for the Advancement of Structured Information Standards (OASIS), (2023)
34. Applebaum, A., Johnson, S., Limiero, M., Smith, M.: Playbook oriented cyber response. In *2018 National Cyber Summit (NCS)*, pages 8–15, (2018)
35. LangChain, Inc. Langchain documentation. https://python.langchain.com/docs, 2025. Accessed: 2025-04-21
36. International Business Machines Corporation (IBM). What is llm orchestration. https://www.ibm.com/think/topics/llm-orchestration, 2024. Accessed: 2025-04-23
37. Paduraru, C., Dumitru, B., Stefanescu, A.: Automated generation of cybersecurity response playbooks via large language models. *Procedia Computer Science*, 270:2987–2996, 2025. 29th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2025)
38. Kucsván, Z.L., Caselli, M., Peter, A., Continella, A.: Inferring recovery steps from cyber threat intelligence reports. In Federico Maggi, Manuel Egele, Mathias Payer, and Michele Carminati, editors, *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 330–349, Cham, (2024). Springer Nature Switzerland
39. Saint-Hilaire, K.A., Neal, C., Cuppens, F., Boulahia-Cuppens, N., Hadji, M.: Optimal automated generation of playbooks. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 191–199. Springer, (2024)
40. Saint-Hilaire, K.A., Neal, C., Cuppens, F., Boulahia-Cuppens, N., Bassi, F., Hadji, M.: A real-time automated attack-defense graph generation approach. Journal of Information Security and Applications **94**, 104266 (2025)
41. Gurabi, M.A., Mandal, A., Popanda, J., Rapp, R., Decker, S.: SASP: a semantic web-based approach for management of sharable cybersecurity playbooks. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–8, (2022)
42. Kremer, R., Wudali, P.N., Momiyama, S., Araki, T., Furukawa, J., Elovici, Y., Shabtai, A.: IC-SECURE: Intelligent System for Assisting Security Experts in Generating Playbooks for Automated Incident Response, (2023)
43. Tsirakis, O., Fysarakis, K., Mavroeidis, V., Papaefstathiou, I.: Operationalizing cybersecurity knowledge: Design, implementation & evaluation of a knowledge management system for cacao playbooks. *Computers & Security*, page 104696, (2025)
44. Mavroeidis, V., Eis, P., Zadnik, M., Caselli, M., Jordan, B.: On the integration of course of action playbooks into shareable cyber threat intelligence. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 2104–2108. IEEE, (2021)
45. Shaked, A., Cherdantseva, Y., Burnap, P.: Model-based incident response playbooks. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–7, (2022)
46. Sang, K.S., Zhou, B.: BPMN security extensions for healthcare process. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pages 2340–2345. IEEE, (2015)
47. Hornsteiner, M., Schönig, S.: SIREN: designing business processes for comprehensive industrial IoT security management. In *International Conference on Design Science Research in Information Systems and Technology*, pages 379–393. Springer, (2023)
48. Granata, D., Rak, M., Salzillo, G., Di Guida, G., Petrillo, S.: Automated threat modelling and risk analysis in e-Government using BPMN. Connect. Sci. **35**(1), 2284645 (2023)
49. Oberhofer, D., Hornsteiner, M., Schönig, S.: Process-Aware Security Standard Compliance Monitoring and Verification for the IIoT. In *2024 European Conference of Information Systems (ECIS24)*, (2024)
50. Zych, M., Mavroeidis, V., Fysarakis, K., Athanatos, M.: Reviewing bpmn as a modeling notation for cacao security playbooks. In *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, pages 490–495. IEEE, (2023)
51. Rahman, M.R., Hezaveh, R.M., Williams, L.: What are the attackers doing now? Automating cyberthreat intelligence extraction from text on pace with the changing threat landscape: A survey. ACM Comput. Surv. **55**(12), 1–36 (2023)
52. Sun, N., Ding, M., Jiang, J., Weikang, X., Mo, X., Tai, Y., Zhang, J.: Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives. IEEE Communications Surveys & Tutorials **25**(3), 1748–1774 (2023)
53. Chaleshtori, F.H., Ray, I.: Automation of vulnerability information extraction using transformer-based language models. In *Computer Security. ESORICS 2022 International Workshops*, pages 645–665, Cham, (2023) Springer International Publishing
54. Neuhaus, S., Zimmermann, T.: Security Trend Analysis with CVE Topic Models. In *2010 IEEE 21st International Symposium on Software Reliability Engineering*, pages 111–120, (2010)
55. Mounika, V., Yuan, X., Bandaru, K.: Analyzing CVE Database Using Unsupervised Topic Modelling. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 72–77, (2019)
56. Li, J., Yadav, A., Sah, A.P., Karim, A.: Enhanced vulnerability severity assessment via graph convolutional network-based dependency analysis. Int. J. Comput. Appl. **48**(1), 96–115 (2026)
57. Park, J.S., Kang, M., Lee, S., Chae, D.-K.: Graph Summarization for Human-Understandable Visualization towards CVE Data Analysis. In *2022 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 314–317, (2022)
58. Sleeman, J., Finin, T., Halem, M.: Understanding cybersecurity threat trends through dynamic topic modeling. Frontiers in big Data **4**, 601529 (2021)
59. Miranda, L., Vieira, D., Pfleger, L., de Aguiar, D., Menasché, S., Bicudo, M.A., Nogueira, M.S., Martins, M., Ventura, L., Senos, L., Lovat, E.: On the flow of software security advisories. IEEE Trans. Netw. Serv. Manage. **18**(2), 1305–1320 (2021)
60. Pydantic community. Pydantic documentation. https://docs.pydantic.dev/latest/, 2025. Accessed: 2025-04-23

61. Górski, T.: Software architecture description in original software publications. Software Impacts **27**, 100802 (2026)
62. Peffers, K., Rothenberger, M., Tuunanen, T., Vaezi, R.: Design science research evaluation. In *Design Science Research in Information Systems. Advances in Theory and Practice: 7th International Conference, DESRIST 2012, Las Vegas, NV, USA, May 14-15, 2012. Proceedings 7*, pages 398–410. Springer, (2012)
63. Cybersecurity and Infrastructure Security Agency (CISA). Cybersecurity Alerts & Advisories, 2025. Accessed: 2025-03-05
64. Van der Aalst, W., Adriansyah, A., Van Dongen, B.: Replaying history on process models for conformance checking and performance analysis. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **2**(2), 182–192 (2012)
65. Liljequist, D., Elfving, B., Roaldsen, K.S.: Intraclass correlation-a discussion and demonstration of basic features. PLoS ONE **14**(7), e0219854 (2019)
66. Cochran, W.G.: *Sampling techniques*. john wiley & sons, (1977)
67. Rozinat, A., Van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. Inf. Syst. **33**(1), 64–95 (2008)