International Journal of Information Security (2024) 23:1215–1230
https://doi.org/10.1007/s10207-023-00760-5

REGULAR CONTRIBUTION

Generating ICS vulnerability playbooks with open standards

Philip Empl[1] · Daniel Schlette[1] · Lukas Stöger[2] · Günther Pernul[1]

Published online: 11 December 2023

Abstract
Organizations face attacks on industrial control systems (ICS) as vulnerabilities are pervasive. However, patching vulnerable systems by simply updating to the newest version is often not an option and shifts focus to workarounds. Beyond pure patching, workarounds specify other remediation measures (e.g., firewall or VPN configuration) that must be taken due to system availability requirements, complexity, or heterogeneous devices. In this paper, we introduce vulnerability playbooks based on open standards. Pushing the envelope of cybersecurity playbooks—steps organizations should follow when responding to cybersecurity incidents reactively—for ICS vulnerability management offers organizations a more transparent, repeatable process and faster, possibly automated actions. We have designed a process model to collect and transform security advisories in *Common Security Advisory Framework* (CSAF) format and generate *Collaborative Automated Course of Action Operations* (CACAO) playbooks based on listed remediation advice. With a proof of concept, we demonstrate that structured CSAF documents can be seamlessly transformed into CACAO playbooks. For our industrial use case, we must also use unstructured security advice highlighting quality differences (compared to CSAF). Our generated 79 standard conformant CACAO playbooks with 485 identified actions hint at imbalanced advice toward patching. Preferably, vendors should include detailed technical remediation advice, provide APIs, and go beyond patching recommendations in their security advisories. Subscribers should structure their assets and use machine learning to normalize, generate, and prioritize CACAO playbooks. With CSAF and CACAO, we see two open standards for handling vulnerabilities.

Keywords Vulnerability playbook · Security advisory · Industrial control system · CSAF · CVRF · CACAO

1 Introduction

Cybersecurity playbooks are about knowing what to do when insecurity becomes apparent. The heavily promoted notion of playbooks captures the description of organizational processes, specified workflows, and individual actions. Security Orchestration, Automation and Response (SOAR) tools rely on playbooks [1], and the US government, special interest groups, and researchers are eager to develop playbooks [2–4]. With industry support, the *Collaborative Automated Course of Action Operations (CACAO)* playbook format aims to standardize playbooks upholding the principle of open standards [5].

Existing playbooks often address incident types (e.g., phishing or malware), and research is focused on incident response [4]. However, using playbooks to handle specific vulnerabilities is another promising field that vulnerability management tools have only partially explored [6]. Industrial Control System (ICS) vulnerability playbooks—steps organizations should follow when dealing with vulnerabilities proactively—can fill the gap and provide additional remediation advice to organizations. We make a first approach to answer the question: Is it possible to generate ICS vulnerability playbooks?

Our work focuses on vulnerability playbooks for ICS and the industrial Internet of Things (IoT). These systems are affected by numerous vulnerabilities and countless attacks. For instance, according to the National Vulner-

✉ Philip Empl
philip.empl@ur.de

Daniel Schlette
daniel.schlette@ur.de

Lukas Stöger
lukas.stoeger@dehn.de

Günther Pernul
guenther.pernul@ur.de

[1] University of Regensburg, Universitätsstr. 31, 93053 Regensburg, Germany

[2] Dehn SE, Hans-Dehn-Straße 1, 92318 Neumarkt in der Oberpfalz, Germany

ability Database (NVD), 72 vulnerabilities for SIMATIC S7 were discovered in the last ten years and caused the vendor to issue patches and security advisories. Moreover, ICS high-availability requirements, complexity, and many heterogeneous devices complicate (manual) vulnerability management and demand measures beyond updating [7]. Thus, ICS vendors typically offer security advisories detailing workarounds for remediation when system availability is a must and patching is not a direct option. In addition, the US Cybersecurity & Infrastructure Security Agency (CISA) maintains a collection of ICS advisories [8].

Looking at security advisories, we see different vendors use different data formats. One such format is the *Common Security Advisory Framework (CSAF)*, an open standard foreseen to exchange machine-readable information [9]. It includes a dedicated section on remediation options which builds the basis for our streamlined, automated vulnerability playbook generation. Organizations can benefit from ICS vulnerability playbooks by reducing the manual handling of workarounds in multiple ways. Most notably, organizations can limit error-prone information extraction and structuring. Automating the process further increases process transparency and data provenance. These improvements are typically associated with playbooks, which leads us to create vulnerability playbooks based on security advisories.

In this work, we design and implement a process model on top of open standards for security advisories (i.e., CSAF) and playbooks (i.e., CACAO) to generate ICS vulnerability playbooks. We aim at demonstrating the practical benefits of structured security advisories making both security advisory publishers and consumers aware of this. In particular, we leverage public advisory sources and preprocess their data. Thereby, we model devices representing Siemens and Cisco assets. In our proof of concept implementation, we query security advisories from two leading ICS vendors and CISA relevant to our use case. In total, we generate 79 vulnerability playbooks and identify 485 workflow actions. Matching terms, which can be customized, are used to classify playbook steps containing the workflow actions. Our main contributions are:

- A process model for generating ICS vulnerability playbooks. The process model covers four phases: (1) querying vulnerability information, (2) sourcing security advisories, (3) converting data in CSAF, and (4) leveraging matching terms to create CACAO playbooks with workflow actions.
- An open-source application[1] to generate vulnerability playbooks with open standards and industry use case.
- Recommendations for improvement and use of security advisory and playbook standards.

The paper is structured as follows. In Sect. 2, we present a motivating ICS vulnerability and the associated security advisory. Additional background on open standards for vulnerabilities, incident response playbooks, and related work is part of Sect. 3. Section 4 details our process model automating the creation of vulnerability playbooks for ICS. Then, we evaluate our approach with a use case and open-source tool implementation in Sect. 5. In Sect. 6, we outline recommendations for better vulnerability handling. In Sect. 7, we conclude with future research directions.

2 Motivation

We illustrate the representation of security advisories with a highly critical (CVSS[2] base score of 10) ICS vulnerability affecting Siemens SIMATIC CP devices, communication processors used in digital factories [10]. Identified by CVE-2022-34819, the vulnerability centers on improper input validation and the resulting heap-based buffer overflow. As a consequence, attackers could execute malicious code and cause production to halt. We use this vulnerability to emphasize aspects of ICS security advisories and their representation in CSAF format.

Figure 1 shows the abbreviated CSAF document. Upfront metadata inform about the CSAF format and the security advisory publisher, typically the vendor of the affected product(s). A string-based title is used to refer to the security advisory. However, product users are mostly interested in security advisories to extract relevant information on vulnerability remediation. Therefore, crucial remediation advice in CSAF is listed inside a remediations array. Besides vendor fixes instructing to update to the newest version (omitted for brevity), Fig. 1 details workarounds as alternative remediation steps. These workarounds help to harden SIMATIC CP devices until patches are installed. In the example CSAF, these include blocking access to a specific port by using an external firewall and disabling a VPN feature.

In the following, we elaborate on data quality issues concerning security advisories and possibilities of CACAO playbooks. Security advisories and (if available) their CSAF documents do not always contain detailed and executable information. The CSAF example in Fig. 1 represents a best-case scenario. Subscribers are faced with security advisories in various data formats, which are often not machine-readable.

Generating CACAO playbooks (Un)structured security advisories, e.g., CSAF, cannot be automatically applied because they have to be put into an executable format

[1] https://www.github.com/ad2play/ad2play.

[2] Common Vulnerability Scoring System (CVSS) is a standardized framework used to assess and communicate the severity of software vulnerabilities.

and mapped to ICS assets. Considering also heterogeneous devices, multiple vulnerabilities, and security advisory sources, automated vulnerability playbook generation is evident. In contrast to manual advisory processing, process consistency can be improved. For instance, the manual vulnerability handling is error-prone or takes even more time. We would like to emphasize that the best-case scenario is not always given. Unfortunately, organizations currently use proprietary data formats to represent playbooks, even though OASIS has published an open playbook standard, CACAO. CACAO promotes standardization and interoperability, allowing seamless integration of different cybersecurity tools. By enabling automation, it reduces response times and manual intervention in vulnerability management. Its human-readable format ensures ease of use and customization, while its integration with various tools streamlines orchestration and automation, ultimately enhancing an organization's overall cybersecurity defense. However, when automatically creating CACAO playbooks from security advisories, we must also deal with unstructured remediation advice until the CSAF standard is established across the industry.

3 Background and related work

Open standards for vulnerability management and incident response playbooks represent foundations for our work. We further discuss related work within this section.

3.1 Open security standards

Vulnerability management relies on a shared understanding of concepts. Open security standards provide the means to cope with low information quality by assisting with uniform representation and content structure. The following standards and data formats are widely recognized in cybersecurity and help organizations handle vulnerabilities.

CVE [11]—Common Vulnerability Enumeration, used to identify and describe vulnerabilities.

CPE [12]—Common Platform Enumeration, used to identify IT/OT assets.

CVSS [13]—Common Vulnerability Scoring System, used to define and assign severity scores.

CVRF/CSAF [14]—Common Vulnerability Reporting Framework/Common Security Advisory Framework, used to describe security advisories.

The open standards and data formats are intended to inform others about vulnerabilities, exploits, and remediation advice [15]. They answer the crucial questions: What characterizes a vulnerability? What systems are affected? How severe is the vulnerability? And what do others need to know about vulnerability remediation?

```
{
  "document": {
    "category": "csaf_security_advisory",
    "csaf_version": 2.0,
    "publisher": {
      "category": "vendor",
      "name": "Siemens ProductCERT"

    },
    "title": "SSA-517377: Multiple
        Vulnerabilities in the SRCS VPN
        Feature in SIMATIC CP Devices"
  },
  "vulnerabilities": [
    {
      "cve": "CVE-2022-34819",
      "remediations": [
        {
          "category":
          "workaround",
          "details": "Block access to port
              5243/udp e.g. with an external
               firewall if possible"
        },
        {
          "category": "workaround",
          "details": "Disable the SINEMA
              Remote Connect Server (SRCS)
              VPN feature"
        }
      ] ...
```

Fig. 1 Excerpt of a CSAF document for CVE-2022-34819 with remediation advice that specifies two workarounds

3.2 Incident response playbooks

Organizations need to define processes, procedures, and actions for incident response. Threat intelligence is also necessary to handle security incidents [16]. Thus, incident response representations with playbooks bridge the gap between processes and data containing both [17]. Mainly two major use cases—the automation of incident response and the sharing of playbooks—have resulted in the development of open standards and data formats (e.g., CACAO, OpenC2, MITRE D3FEND, or RE&CT) for playbooks and individual actions [18–20].

CACAO. Collaboratively developed by the Organization for the Advancement of Structured Information Standards (OASIS) and its members, the open CACAO format targets playbooks. In contrast to more action-focused standards, CACAO playbooks can describe information on various granularity levels. As a result, the CACAO format is comprehensive and a promising candidate for the description of vulnerability playbooks. To the best of our knowledge, there are no other maintained and open playbook standards with similar characteristics. Using CACAO playbooks, organizations can follow defined workflows and have the ability to automate repetitive, error-prone tasks.

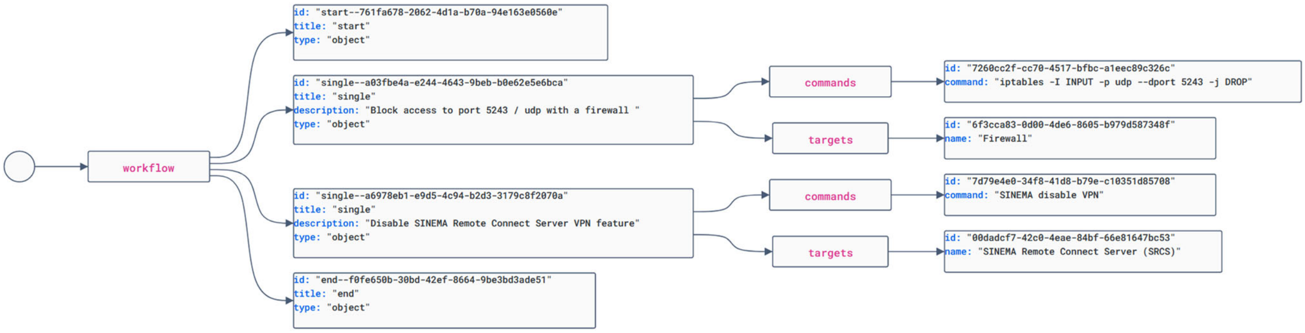

workflow
id: "start--761fa678-2062-4d1a-b70a-94e163e0560e"
title: "start"
type: "object"
id: "single--a03fbe4a-e244-4643-9beb-b0e62e5e6bca"
title: "single"
description: "Block access to port 5243 / udp with a firewall "
type: "object"
id: "single--a6978eb1-e9d5-4c94-b2d3-3179c8f2070a"
title: "single"
description: "Disable SINEMA Remote Connect Server VPN feature"
type: "object"
id: "end--f0fe650b-30bd-42ef-8664-9be3bd3ade51"
title: "end"
type: "object"
commands
targets
commands
targets
id: "7260cc2f-cc70-4517-bfbc-a1eec89c326c"
command: "iptables -I INPUT -p udp --dport 5243 -j DROP"
id: "6f3cca83-0d00-4de6-8605-b979d587348f"
name: "Firewall"
id: "7d79e4e0-34f8-41d8-b79e-c10351d85708"
command: "SINEMA disable VPN"
id: "00dadcf7-42c0-4eae-84bf-66e81647bc53"
name: "SINEMA Remote Connect Server (SRCS)"

Fig. 2 Schematic visualization of a CACAO vulnerability playbook that includes workflow, command, and target objects

The benefits of the CACAO playbook format are best understood by looking at its structure and object types. Figure 2 shows the visualization of a vulnerability playbook for CVE-2022-34819 and the underlying attribute–value pairs in JSON. The playbook is based on real-world vendor remediation advice augmented with commands. CACAO playbooks contain workflows to structure workflow steps. Typically, start and end steps enclose single action steps outlining specific actions. On a more granular level, command and target objects describe executable information and its recipients. In the example, organizations can derive two remediation actions, systems involved (i.e., firewall, server), and commands (i.e., iptables, disable). CACAO is broad in scope, and command and target types also support manual actions for individuals. Adding conditional workflow steps helps to represent sophisticated workflows. We use CACAO as it can capture multiple CSAF-based remediation measures and hand action-based workflows to organizations. In the remainder of this paper, we refer to CACAO workflow steps as workflow actions to differentiate between CSAF and CACAO.

3.3 Related work

Vulnerabilities and vulnerability management are of interest to researchers and organizations alike. Organizations are advised to systematically handle vulnerabilities as they can lead to severe security incidents [21]. A steady stream of research covers general and ICS-specific vulnerabilities [22, 23]. From a management perspective, CISA provides a so-called vulnerability response playbook to assist organizations in deciding about vulnerability handling [6]. From vulnerability discovery [24], to vulnerability assessment [25] and security advisories [26], transparent processes and standards are important. While Fenz et al. [27] introduce automated handling of security advisories, other work has taken on the challenge to provide commit-level patch advice for vulnerabilities [28]. Besides, vulnerability management is of practical interest as vendors of commercial vulnerability management tools address the need to keep track of assets and vulnerabilities.

Academic work on cybersecurity playbooks is sparse. Nevertheless, playbooks are an emerging research topic related to threat intelligence and security standards [17]. In a recent study, Stevens et al. [4] explored human playbook creation with available frameworks indicating playbook variety. As different approaches and sharing use cases exist, integration and semantics of playbooks are investigated [29–31]. Against the backdrop of security orchestration and a plethora of commercial SOAR tools [1], specific use cases (e.g., an IoT context with digital twins) have been discussed [32, 33]. It can be seen that playbook generation is crucial to leverage SOAR tools.

We go beyond related work in the following ways. Our approach is the first to combine the two areas of security advisories and playbooks. Building vulnerability playbooks offers organizations more process-oriented advice on what to do. While some vulnerability management tools incorporate the idea of playbooks, we see benefits in following a similar path with open security standards. Our focus on ICS security advisories capitalizes on the fact that remediation measures are most important when simply patching is not an option. Playbooks can introduce transparent processes and automation toward better vulnerability management for ICS.

4 Vulnerability playbook generation

Driven by the problem of ICS vulnerability handling and inspired by related works, we develop a process model. Our approach follows the design science research methodology [34] starting with a problem and developing an artifact to be evaluated and communicated. Our artifact is a process model that aims for a complete output (playbooks). From *security advisory to vulnerability playbook*, the process captures automated ICS vulnerability playbook generation with four phases shown as a Business Process Model and Notation (BPMN) diagram in Fig. 3. The subsequent sections are dedicated to the illustrated process phases.

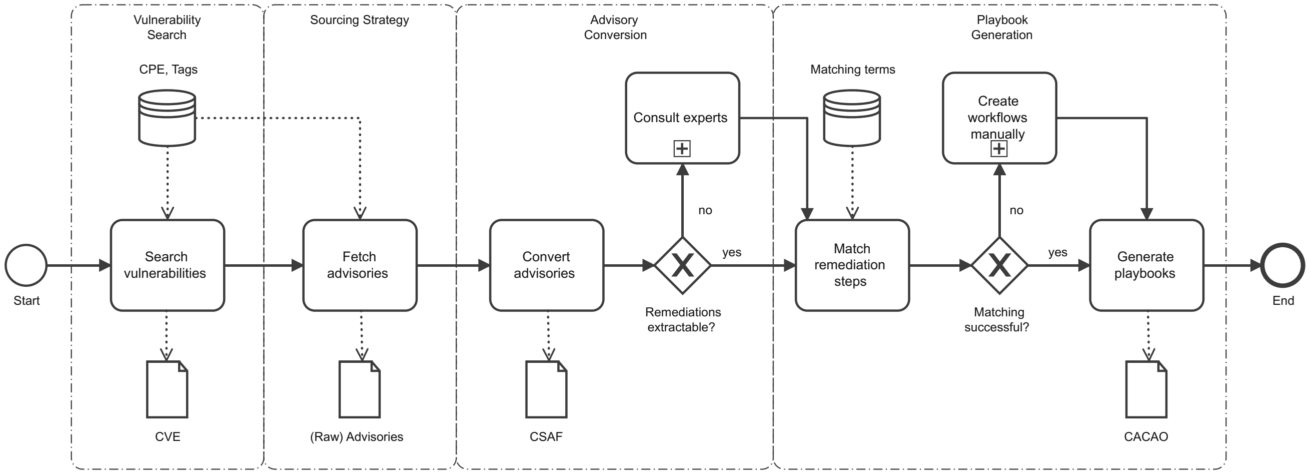


Fig. 3 Process description from security advisory to vulnerability playbook

4.1 Vulnerability search

Vulnerability handling and the playbook generation process start with assets and the question of whether these assets are vulnerable or not. Thus, we define the activity *Search vulnerabilities* to get an overview of relevant vulnerabilities. As a prerequisite, organizations must already carefully document their assets and components (e.g., virtual representations or SBOM—Software Bill of Materials). Using this documentation, assets and respective identifiers (e.g., CPE-ID) are used to find vulnerabilities. However, the specific characteristics of ICS need to be considered. Most notably, ICS assets are built of multiple components forming complex systems-of-systems [35]. Each of the components can run its own software on dedicated hardware. Searching for relevant security vulnerabilities requires identifiers—for the vulnerability and the components. Vulnerabilities are given a CVE-ID. ICS components (i.e., hardware or software) have a CPE-ID or other tags. If a component is described by CPE, querying associated CVEs is straightforward. Without CPE, other information (e.g., model or version) must be used to search vulnerability databases. In our process model intended for automation, we rely on CPE or, when not available, use device-specific tags. Both approaches enable automated vulnerability searches, but using tags might lead to more errors. The first phase yields vulnerabilities regardless of available security advisories.

4.2 Sourcing strategy

The activity *Fetch advisories* is part of the second process phase. Our sourcing strategy involves security advisory acquisition from product vendors. These product vendors often have Product Security Incident Response Teams (PSIRTs/ProductCERTs) that offer vulnerability remediation advice for their products. In addition, other institutions (e.g., national or coordination Computer Emergency Response Teams, CERTs/CSIRTs) and commercial security vendors partially aggregate security advice. In most cases, security advisories can be fetched with CPEs or tags and link to CVEs. Focusing on ICS and fetching security advisories for systems-of-systems involves multiple sources varying in format, structure, and content. We compared security advisory publishers and data formats. Most sources provide access to PDF security advisories or embed these directly on their website. In the best-case scenario, we find dedicated formats such as CSAF or its predecessor CVRF, but they can be retrieved less often. Formats like HTML or PDF require deep traversal and scraping. Since both formats do not provide options to directly map assets to remediation advice, we need to filter whether the remediation advice actually affects devices of interest. Therefore, we recommend the device representations to skim and filter these documents automatically. As a means of communication, many of the listed sources offer RSS feeds, email notifications, or communicate the latest advice via Twitter.

The various communication channels do not solve the problem of data heterogeneity and do not always allow the exchange of remediation advice and feedback. Additionally, many sources do not provide an API to fetch security advisories for specific vulnerabilities. Automating the filtering of RSS feeds or emails to match the advisories of interest is an unnecessarily complex intermediate step. Organizations relying on different sources and advisory formats must convert and standardize these advisories to enable automation.

4.3 Advisory conversion

The activity *Convert advisories* targets standardization and the results are uniform security advisories. Since different

Table 1 Action classification and related terms based on the OpenC2 commands

Class	Terms
Update	[["patch", "update", "upgrade"], ["version", "v", "ver"]]
Investigation	[["investigation", "investigate", "scan", "examine", "inspect", "inspection", "review", "check"]]
Locating	[["locate", "find", "detect", "discover", "uncover"], ["object", "artifact", "file", "directory", "instance"]]
Data operation	[["query", "create", "alter", "delete", "copy"], ["data", "entity", "directory", "file"]]
Isolation	[["contain", "containment", "isolation", "avoid"], ["file", "process", "entity", "asset"]]
Privileges	[["access", "credentials", "right"], ["allow", "restrict", "grant", "assign", "give", "permit", "reduce", "regulate", "block", "limit"]]
System	[["start", "stop", "restart", "cancel", "enable", "disable"], ["process", "application", "system", "activity", "action", "environment", "function", "port"]]
Configuration	[["set", "change", "apply", "put", "restore"], ["value", "configuration", "state", "property", "attribute"]]
Network	[["redirect", "switch", "block", "intercept"], ["traffic", "destination", "url", "ip", "port", "address", "packet", "network"]]
Observation	[["detonate", "execute", "observe", "examine", "monitor", "discover"], ["behaviour", "malware", "target", "action", "attack", "activity"]]

vendors use different formats for representing and sharing security advisories, it is essential to convert these heterogeneous security advisories into a uniform format before generating playbooks. We rely on the open standard CSAF for the structuring and presentation of remediation steps. Thus, it is the objective of this activity to convert security advisories into CSAF documents.

There are three possible cases. First, CVRF is converted to CSAF using semantically identical fields to store remediation steps. Second, when security advisories are provided as CSAF documents, they are not converted and taken as is. However, we discard remediation steps not matching the CPE identifiers or tags. Last, also other source-specific types of security advisory formats are converted. Here, remediation advice needs to be extracted. Dependent on the data format, steps may include HTML/PDF parsing and scraping to identify and extract nested remediation steps. For instance, in the case of CISA security advisories, we suggest to extract the mitigation section, the executive summary, and the technical details besides relevant metadata, i.e., title, date, or URL. Note that these steps require logic to filter the remediation advice as unstructured data do not maintain a reliable mapping between remediation advice and devices of interest. In a scenario where no remediation advice is available, we include user interaction and consult experts. Possible calls to action include the search for internal playbooks targeting similar vulnerabilities. These playbooks might provide remediation steps that can fit the currently investigated vulnerability. Regardless of the scenario, this process phase results in standardized security advisories with remediation steps for vulnerability playbook generation.

4.4 Playbook generation

After unifying security advisories, we move from security advisories to playbooks involving activities to *Match remediation steps* and *Generate playbooks*. We rely on the open standard CACAO for structuring playbook-related information and workflow actions. In CSAF, remediation steps are mostly textual descriptions that are not actionable. We aim at deriving workflow actions for playbook execution. Thereby, we take remediation steps from CSAF, classify them, and put appropriate predefined actions into the CACAO workflow action section.

We introduce the concept of *matching terms* deciding about the class of a specific workflow action. In advance, organizations must define and assign action templates to specific classes. This allows playbooks to be dynamically populated with respective actions matching a class. These matching terms resemble a two-dimensional search on remediation steps. One dimension describes the action and the other dimension the target. Matching both dimensions is essential to meet a stemmed matching term fully. A given string must at least match one word per dimension to reach the next one. With the approach, it is possible to define complex matching terms. We initially define the action classes based on individual actions (e.g., create, update, or delete) proposed by the OASIS OpenC2 standard. Note that organizations are flexible in their choice of classification, mapping, and creation of specific action templates; OpenC2 is only one option to classify. The classification helps to tag the playbook accordingly. While these actions are used in our work to classify workflow steps, CACAO command objects can capture OpenC2 commands supporting agnostic automation. Table 1 shows possible action classes and related matching terms. For full automation, organizations must create and assign action templates to specific classes and matching terms to populate the playbook dynamically. These action templates might be selected based on the matched terms and dynamically fed by variables (e.g., port = 5243 and target = firewall).

Applying matching terms to remediation steps requires disassembling these steps into sentences and understanding their intention. Natural language processing (NLP) is an accepted method to process and understand human-readable language. Breaking a remediation step into sentences and tokenizing each sentence lead to a set of words. Then, these words are brought into the basic form using stemming. Finally, the action class is identified if a stemmed matching term applies to a sentence across its dimensions. The following example demonstrates the two-dimensional mapping matching the terms "block" and "port" and resulting in the "network" action class:

Remediation step: *block access to port 5243/udp*
→ Stemming: *[block, access, to, port, 5243/udp]*
→ Matching: *[[block],[port]]*
→ Tag: *Class* → *network*
Suggested action: *Block port (port: 5243, target: firewall)*

As can be seen, the matching terms identify the class of a workflow action. Playbook-relevant parameters can be passed in this context. Ideally, actions should rely on predefined commands fitting match term combinations to automate the vulnerability handling completely. Toward automated execution of vulnerability playbooks, more granular action classes with more matching term combinations are necessary. Nevertheless, workflow steps are only one part of a CACAO playbook. Besides the workflow, a CACAO playbook also contains metadata and targets. The playbook generation activity places the remediation steps in the workflow section of the CACAO playbook and fills the remaining fields with metadata and additional information.

Our process model tends not to automate the whole process, from identifying a vulnerability to its remediation. We see this process model as a means to assist analysts by identifying and suggesting asset-relevant security advice. The playbook generation phase also involves two manual steps. First, if there is a matching error, e.g., no classification is possible, analysts can manually label workflow actions to continue the process. Second, the process model ends after suggesting a vulnerability playbook to the analyst. It is then up to the analysts whether they would like to execute, adjust, or delete the playbook. Of course, in a best-case scenario, these steps would be automated, although it is questionable whether organizations are willed to apply remediation advice to critical assets without reviewing them.

5 Evaluation

We show that it is feasible to seamlessly generate vulnerability playbooks from structured security advisories with a reasonable amount of effort. Additionally, we compare the quality and completeness of playbooks generated using structured and unstructured security advisories. In doing so, we implement our process model with a proof of concept satisfying a real-world industrial use case. Our use case defines two device representations to model systems-of-systems with vendor-specific components. Our application implements the *security advisory to vulnerability playbook* process aggregating remediation advice from three sources differing in data format, namely *Siemens ProductCERT*—CSAF, *Cisco Security Advisories*—CVRF, and *CISA ICS CERT*—HTML.

5.1 Industrial use case

Our real-world industrial use case describes an enterprise, namely Dehn SE, that is a market leader in plant and building technology, traffic and telecommunications systems, the process industry, and photovoltaic and wind power plants. As a manufacturing enterprise with over 2000 employees, the ICS consists of several assets from Siemens and Cisco. The enterprise already tracks the vulnerabilities of IT assets, such as software packages. The monitoring of vulnerabilities in the ICS is currently still under development. Tracking vulnerabilities and managing remediation advice is perceived as a mammoth task due to the heterogeneity and plethora of assets in use. The enterprise is highly interested in an automated solution gathering vulnerabilities and remediation advice for its assets.

In order to track their ICS assets, we model two virtual representations (i.e., components, CPE identifier, and tags) detailing ICS assets in use. These representations aggregate assets by vendor, thereby forming complex systems-of-systems. To not reveal the actual assets in use, we have augmented them with several other products of the respective vendor. In doing so, we created two obfuscated representations. Of course, other use cases may have other system representations. The first system comprises 22 Siemens field devices, e.g., Siemens SIMATIC S7 (see Appendix 1), typically used in industrial automation and control systems. The second system defines 17 Cisco networking devices, e.g., used as gateways or controllers, found in ICS networks.

5.2 Experimental setting

Our experimental setting consists of adequate hardware and software serving the industrial use case. We have implemented an application with a user interface to efficiently integrate analysts into the vulnerability playbook generation process.

We run all experiments on a single virtual machine with Ubuntu 22.04 LTS operating system, 8GB RAM, and 80GB storage. The device representations are structured using JSON, similar to the widely used Eclipse Ditto[3] representation. The application is based on a front-end/back-end architecture and fully conforms to the CSAF and CACAO standards. The front end is based on Vue.js and the back end on Node.js. The front end is the entry point for the user to verify the correct processing of the security advisories. It provides several functions: CSAF and CACAO visualization, task overview and execution, matching term management, a CSAF converter, and a playbook configurator. A task[4] is considered open if no workflow actions can be derived. A task is done when the workflow actions have been successfully processed, but the final human assessment and approval are pending. The back end relies on the model–view–controller principle and stores CACAO and CSAF documents in a MongoDB. We provide a dashboard for all tasks and their states. Additionally, an analyst can manage the device representations and integrated sources. The pattern section is dedicated to the definition of matching terms.

Our evaluation is threefold. We first run the application, gathering the security advisories (input) to generate playbooks (output). Afterward, we manually assess the input and compare it with the output to assess the overall playbook quality and completeness. As input, we rely on security advisories from different sources for the respective devices. Therefore, we have integrated security advisories from three sources: Siemens ProductCERT, Cisco PSIRT, and CISA ICS CERT. Our application automatically fetches remediation advice from these sources and prevents us from fetching the same advisories multiple times. We selected these sources as they offer vendor-specific or aggregated security advice. Second, these sources ultimately use different data formats to evaluate whether structured security advisories lead to more qualitative and complete playbooks. We collected security advisories over the last 150 days for the playbook quality evaluation. As we also had to assess the security advisories manually, we considered only a collection period of 150 days, although our application could fetch and process even more advisories. After these advisories passed the whole process, we compare the following key indicators to evaluate the playbook's quality and completeness:

- Quantity of workflows actions
- Mistaken acceptance of workflow actions (*type I error*)
- Mistaken rejection of workflow actions (*type II error*)
- Classification of workflow actions

Third, we evaluate the performance of our automated process model showing that automation changes the game in managing vulnerabilities for ICS assets. For the performance measurement, we collect security advisories targeting our assets from the last five years. Through the manual labeling process, the human assessment, and performance measurements, our experimental setting led to several results.

5.3 Experimental results

We have grouped our results according to the process phases from security advisory to vulnerability playbook. Additionally, we show results concerning playbook quality, completeness, and performance. The results are documented using a Jupyter notebook to create transparency, which is available on GitHub.[5]

Vulnerability search. In the industrial use case, device representations hold asset information, including CPE-IDs. We noticed that we could not assign a CPE-ID to each component. This problem has also been pointed out by previous research [36]. We found 13 CPE-IDs for the 17 Cisco assets and 22 CPE-IDs for the 20 Siemens assets. At first glance, these numbers sound reasonable, but considering that CPE can address assets' firmware and hardware, we expected 34 and 40 CPE-IDs, respectively. In addition to the CPE-IDs, we added device-specific tags (i.e., model number). We found 35 vulnerabilities for our devices. Grasping the insecurity of ICS with these asset-specific vulnerabilities, we follow up with the search for security advisories.

Sourcing strategy. Integrating the security advisory sources was a significant challenge due to their heterogeneity. The Siemens ProductCERT does not provide an API. Instead, they offer an Atom feed to query CSAF security advisories using the SSA ID[6], CVE, title, product, sector, or tags. We use the advisory identifier within the Atom feed to manipulate the Siemens website URL and request the advisory in CSAF format. The Cisco PSIRT provides an API based on open security standards (e.g., CVE, CVSS, and CVRF).[7] Since the API does only respond with XML-based CVRF, we still need to convert it. Finally, the CISA ICS CERT does not provide an API or feed to retrieve security advisories. Using the device tags of the device representations, we search within the HTML document and scrape information from its remediation section. As can be seen, searching for remediation advice without any interface and filtering options is a fundamental problem. Therefore, we had to use the device tags and CPEs to automate filtering and verify

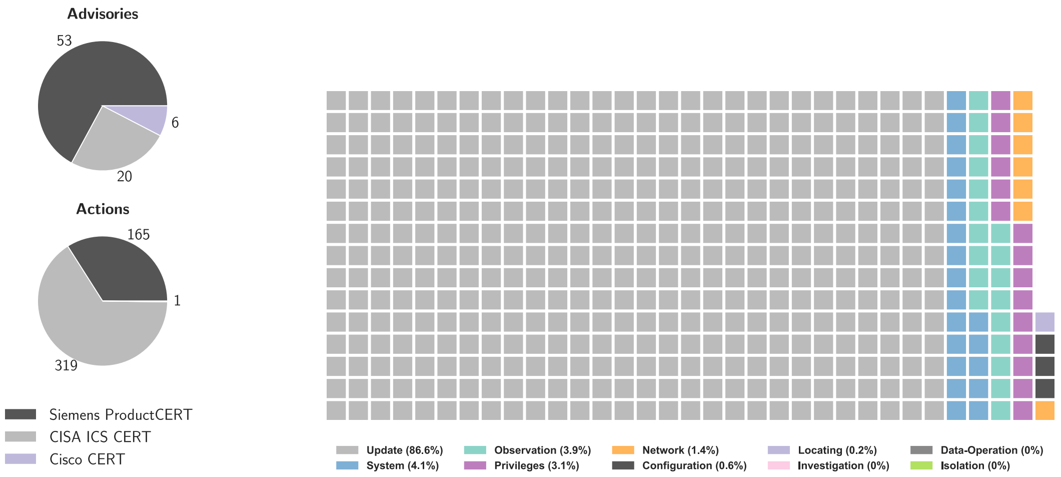
[3] https://www.eclipse.org/ditto/.

[4] A task manages the generation of one playbook.

[5] https://github.com/ad2play/evaluation.

[6] Siemens Security Advisory (SSA) is a Siemens global security advisory identifier.

[7] We noticed that the Cisco OpenVuln API recently added support for CSAF documents.



(a) 79 Advisories and 485 actions by CERT.

(b) Classification of 485 identified workflow actions.

Fig. 4 Analyzing the workflow actions in the generated CACAO playbooks

whether the security advisory is associated with the asset and the respective vulnerability. Since they categorize vulnerabilities, products, and remediation steps, filtering is only a minor problem within CSAF/CVRF documents.

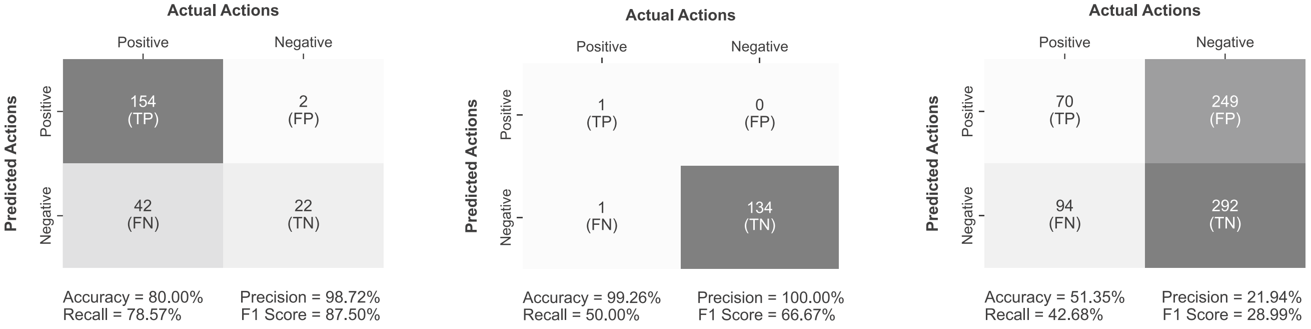
We identified 79 security advisories (see Fig. 4a). Siemens offers 53 advisories, and Cisco offers six. CISA usually lists security advisories for both Siemens and Cisco devices, but the CISA advisories that have been fetched do not contain remediation advice for the Cisco device. However, Cisco has generally listed fewer advisories in the period in question. Also, CISA ICS CERT advisories primarily focus on ICS and do not cover Cisco products for IT enterprise networks. CISA provides a total of 20 security advisories for Siemens assets. It is also noticeable that Siemens offers several versions of advisories, but most overlap considerably in content. Therefore, the total number of Siemens advisories is significantly higher than those from CISA. In addition to the three sources mentioned above, we skimmed IBM X-Force Exchange and NVD [37][8] for security advisories. There, we could find remediation advice only in linked external vendor documents creating complexity for our use case. At the end, we notice that different sources imply different obstacles in obtaining security advisories for specific assets, making sourcing inconvenient.

Advisory conversion. After successfully acquiring security advisories, they are automatically converted into the CSAF data format. For Siemens advisories, already available in CSAF format, no further steps are necessary. The security advisories from Cisco and CISA are converted into CSAF using CVRF and HTML adapters, respectively. When the security advisories from all sources have been converted to CSAF, we analyze these documents.

A closer look at the remediation steps leading to workflow actions (see Fig. 4a) also shows that the amount varies by vendor. While CISA has 319 remediation steps in 20 advisories (16 steps per advisory), Siemens captures 165 remediation steps (3 steps per advisory), and Cisco provides only one remediation step. The identified number of workflow steps in CISA might indicate a high type I error, but it is noticeable that CISA offers additional remediation advice compared to vendor-specific ones. Most interestingly, vendors even advertise remediation advice to inform customers that there is currently no fix available. None of the vendors directly offers technical commands (e.g., in OpenC2 or else) in the remediation steps, whereby dealing with textual descriptions of remediation advice is crucial. In conclusion, advisory conversion is strongly action-centric identifying individual remediation steps.

Playbook generation. The standardized security advisories in CSAF enable the generation of CACAO playbooks. CACAO is an extensive standard, and its implementation is challenging. Emblematic for this fact, the generated CACAO playbooks have a total length of 29,100 lines of code, which leads to 410 lines per playbook. Appendix 1 shows an excerpt of a generated CACAO playbook. However, generated CACAO playbooks are shorter than the initial CSAF documents. One reason is that CSAF also lists remediation advice for other assets, which were not required for our industrial use case. We successfully generated 71 CACAO playbooks out of 79 CSAF documents. Eight advisories require manual post-processing as actions could not be classified correctly. These eight reworks can be traced back to two issues. Seven errors are due to the NLP procedure, which has problems processing placeholders in version numbers, such

[8] NVD by NIST is a comprehensive repository of information related to publicly known cybersecurity vulnerabilities.



(a) Siemens ProductCERT ($n = 220$).

(b) Cisco PSIRT ($n = 136$).

(c) CISA ICS CERT ($n = 705$).

Fig. 5 Measuring the CACAO playbook quality along workflow action classes (n equals the number of workflow actions)

as "update to version 3.X." The other error occurred because one remediation step could potentially be assigned to two different classes. Still, we can reduce the manual effort by roughly 90% and automating remediation advice can be seen as a success. Of course, final human assessment is crucial to performing the correct workflow actions to the right target at the right time.

Another elementary part of the CACAO playbook generation is the classification of the individual workflow actions (see Fig. 4b). It is striking that 86.6% of the workflow actions force an update, whereas system (4.1%), observation (3.9%), and access (3.1%) play a rather subordinate role. We also found that the class observation is only mentioned in CISA security advisories. They have a dedicated section advising to observe malicious activity and to report security incidents. The lack of contextual understanding is also a problem while using NLP. These above numbers are the output we yield within the automated process. Ensuring that the generated playbooks match the security advisories' content requires determining the overall CACAO playbook quality and completeness.

Playbook quality and completeness. We have already seen that the automated creation of CACAO playbooks is feasible and promising. We evaluate the extent to which these results are actually correct in the following. We measure the playbook quality and completeness by referring to confusion matrices (see Fig. 5). These confusion matrices shows the three sources and an overall estimation of the playbooks' quality and completeness. We thereby include the correct amount of actions and their classification. We calculate the type I error as falsely identified workflow actions. The type II error represents the incorrectly rejected workflow actions. The total number of potential actions is given by the total number of sentences in the remediation steps ($= n$) because, except for one remediation step, all workflow actions were assigned unambiguously to a specific class. We assume that each workflow action is targeted by one sentence. Figure 5a shows Siemens security advisories' precision, accuracy, recall, and $F1$ score. The high precision (98.72%) shows a high quality of the generated CACAO playbooks. This indicates that the playbook quality is kept high when vendors provide security advisories in CSAF format. Only relevant remediation advice is included in the playbook generation process, while insignificant workflow actions are disregarded. The recall of 78.57% shows acceptable completeness of workflow actions indicating that only a small proportion of workflow actions is actually missing within the playbook. The CACAO playbook generation ($F1$ score = 87.5%) using structured and machine-readable security advisories is outstanding. The type I error is 1%, and the type II error is 19%, which signifies that the matching terms may be too soft. For example, the locating and isolation classes have been matched several times on the first dimension, but did not succeed on dimension two. Figure 5b portrays the results for the Cisco PSIRT using the structured CSAF predecessor CVRF. This leads to an averaged result with an $F1$ score of 66.67%, a type I error of 0%, and a type II error of 0.007%. These results are insignificant, but we decided to include them for completeness. In contrast, unstructured security advisories from CISA deliver different stats (see Fig. 5c). The generated playbooks for CISA are qualitatively inferior compared to Siemens, which is reflected by a low precision of 21.94%. The identified workflow actions show higher incompleteness (precision = 42.68%), leading to an $F1$ score of 28.99%. The type I error is 35.3%, and the type II error is 13.3%. The direct comparison reveals that clear structured, machine-readable security advisories lead to more qualitative and complete playbooks, which in turn results in fewer manual corrections. The matching terms are an adjustment screw to balance the type I and type II errors, but the quality of the fetched security advisories is decisive.

Performance. We have found that fully automating the process, starting with vulnerability search and ending with playbook generation, saves time and reduces effort. For measuring the performance, we use the experimental setting mentioned above. We have collected vulnerabilities and secu-

Table 2 Performance of each process phase

	Vulnerability search	Sourcing strategy	Advisory conversion	Playbook generation
∅	1.6 s/CVE	0.06 s/adv	0.03 s/CSAF	0.06 s/CACAO
$\sum$	7.42 min	27.23 s	14.38 s	20.74 s

rity advisories for our devices for the last 5 years (as of December 2022). Table 2 shows each process phase's average/total duration, respectively. It takes 7.42 min to lookup and filter vulnerabilities for 35 CPE-IDs and device tags (3784 unfiltered; 266 filtered). As components are not always mapped to a specific CPE-ID, our tool also performs searches with device tags. Due to the exhaustive filtering, we find long runs during vulnerability searches. Afterward, the tool uses this input to fetch 440 security advisories from different sources (Cisco: 112, Siemens: 267, CISA: 61), which takes 27.23 s (Cisco: 5.55 s, Siemens: 11.08 s, CISA: 10.6 s). Siemens advisory sourcing takes twice as long because two different API calls are required; the first API call fetches the RSS feed, and the second downloads respective advisories. Advisory conversion takes 14.38 s (Cisco: 14.05, Siemens: 0 s, CISA: 0.33 s). As we use the dedicated Cisco API to transform CVRF to CSAF, these operations take longer. The automation successfully maps and generates 323 playbooks out of 440 advisories from these advisories in 20.74 s. In summary, using five years of historical data, it takes 8.46 min to automatically generate playbooks for our devices. We observe 1.57 s on average to progress all process phases identifying a component's vulnerability, deriving appropriate remediation, and generating a playbook. It is up to organizations to develop runtime (performance) optimization strategies and achieve higher scalability for complex environments. For instance, by increasing CPUs or caching results, fetching and processing security advisories should be more efficient.

5.4 Limitations

We have a few limitations concerning the application and evaluation. Design decisions had to be made in implementing our application following our process model. Therefore, we extended the JSON schema of CACAO and CSAF to a small extent due to the choice of specific technologies. For example, in the CACAO schema, we had to exclude trailing dollar signs for the data type identifiers to maintain compatibility with MongoDB. In addition, the proposed NLP procedure is inaccurate in terms of contextual understanding, the distinction between nouns and verbs, or sketchy texts. Our NLP implementation cannot accurately pinpoint the relationship between actions and targets. Additionally, we cannot identify the target. In addition, our evaluation has some further limitations. First, it is partway biased due to a large number of security advisories from Siemens ProductCERT and CISA ICS Cert. Hence, we cannot generally argue about the generated playbooks' quality and completeness across all security advisories. We can only observe that structured data yield better results than unstructured. Second, we have only connected three CERTs as potential sources for security advisories (limited to the last 150 days) based on our devices. And third, our playbook generation does not retain conditional logic or parallel flows (if existent in security advisories). The current mapping is rigidly sequential. We declare the handling of different versions of security advisories out of scope, e.g., those from Siemens ProductCERT. Last, we face limitations regarding the dependence on physical processes, insufficient contextual knowledge, limitations in dealing with hardware modifications, complex configuration and documentation requirements, applicability to small environments, and modeling temporary response actions.

6 Recommendations

We summarize the results and present recommendations for publishing security advisories directed at CERTs (*advisory publishers*) and automating ICS vulnerability handling directed at asset owners (*advisory subscribers*). The latter strongly depends on whether the publisher already provides ambitious remediation advice. Otherwise, subscribers have to assemble ambiguous remediation advice.

6.1 Publishing security advisories

We see a remarkable improvement potential for exchanging security advisories on the publishers' side. Publishers (i.e., vendors and other CERTs) should enable more automated remediation advice retrieval for subscribers and foster a standardized exchange of security advisories.

Enable automated advice retrieval. We have found that many data formats currently create a massive information overhead and expenses for subscribers of security advisories. One reason is that publishers only offer traditional communication channels, such as RSS feeds or email notifications. For a targeted query of relevant security advisories and to avoid information overhead, it is of utmost importance to offer a standardized API that additionally provides customization, i.e., filtering options. APIs should leave it to the subscribers which data format they prefer for their remediation advice. This would make searching for security advisories less painful and more efficient. Additionally, API access allows security advisories to be retrieved in real time or near real time, ensuring that subscribers receive the most current information about vulnerabilities. This is crucial for promptly addressing potential security risks. Beside playbook generation, APIs enable seamless integration of security advisory data into various systems, applications, and tools (e.g., SOAR) used by security professionals. This integration facilitates automated processes for vulnerability scanning, patch management, and incident response, reducing manual effort and potential human errors. Last, APIs are designed to handle high volumes of requests, making them suitable for distributing security advisories to a large number of subscribers and systems efficiently. Overall, providing CSAF-structured security advisories via API fosters a more efficient, interconnected, and responsive cybersecurity ecosystem, enabling organizations to stay proactive and better defend against emerging threats.

Use structured security advisories. Publishers should offer structured security advisories making the content easily machine-readable. Most data formats (i.e., HTML or PDF) for exchanging security advisories differ in structure and content. We have found that structured data formats (i.e., CSAF) better support automation than unstructured data by providing dedicated sections for actions and targets and tend to be more machine-readable. Translating unstructured data into machine-readable advisories requires sophisticated techniques coined by errors. In addition, structured data simplify uniform handling without striving for different conversions of the security advisories. We also came across some best practices for the security advisories' content. First, publishers should only include relevant information in security advisories to keep the remediation advice clean and to prevent information overhead. We propose to structure advisories using the actuator–action–artifact triplet [17]. This triplet helps organize information about the actuator (e.g., firewall), action (e.g., blocking), and artifact (e.g., IP address), normalizing the content of advisories. Second, publishers should be aware of streamlining, maintaining, and optimizing remediation advice. We believe versioning of security advisories to be helpful, as additional remediation advice extends to newly affected assets while keeping the total quantity of security advisories the same. Next, publishers should dedicate a sentence to each remediation step to foster automation. Additionally, as updates are not always feasible, publishers should include more "real" workarounds. Ideally, publishers should keep the CVE and product identifiers within security advisories. It can be observed that some security advisories do not list CVE-IDs. However, there is different remediation advice for different products and publishers should continue mapping product identifiers to individual remediation steps. If this mapping is missing, subscribers cannot ensure that the remediation advice is meant for their assets. Last, we recommend publishing asset-specific commands, needed for automated playbook execution. For that purpose, CACAO defines command types that can capture OpenC2 commands.

6.2 Automating vulnerability handling

Automating vulnerability handling is crucial to cope with the increased number of threats. We summarize our key learning and provide recommendations for security advisory subscribers. Structured device representations, a clear prioritization and sourcing strategy, the integration of machine learning, and the adoption of CACAO playbooks are enablers for automation.

Use structured device representation. Subscribers must know their devices, components (including versions), and vulnerabilities. Organizations need to keep track of hardware modifications and require configuration and documentation management. Comprehensive, well-structured, integrated device representations are the cornerstone for identifying and automating relevant remediation advice. We recommend using a structured format (e.g., JSON or SBOM) and device representations to model complex systems-of-systems. Enriching and maintaining these representations with security-relevant information (e.g., CPE) is crucial to identify vulnerabilities, exploits, and remediation advice.

Integrate machine learning. Subscribers should pay particular attention when selecting appropriate security advisory sources. As these sources differ in many aspects, subscribers have to decide whether the added value of a potential source outweighs the effort involved. The effort usually results from the additional development for security advisories' conversion. For high quality, subscribers should directly integrate vendor-specific advisory sources if they plan automated processing. Free-to-use sources that aggregate remediation advice (e.g., CISA ICS CERT) list advisories from several vendors but are less suitable for automation. Alternatively, subscribers can obtain aggregated security advisories from security vendors without worrying about integrating different vendors.

Integrate machine learning. The integration of machine learning for the automated identification of actions and targets is promising. As long as some CERTs advertise remediation steps in plain text, subscribers should consider whether the application of machine learning can lead to a general improvement in automation. Sophisticated machine learning techniques could lead to sounder contextual understanding and, thus, better automation, quality, and completeness of workflow actions. In particular, large language models hold tremendous potential. With their advanced natural language processing capabilities, large language models can efficiently analyze and interpret textual information to identify appropriate matches and classifications of actuators, actions, and artifacts. In detail, large language models can better capture the subject, verb, and object of a sentence in order to address the current heterogeneity of different data formats for security advisories. In addition, these models can identify commands and modify them to fit within an organization's landscape. When they incorporate organizational knowledge through embedding, they may predict the relationship between assets (actuators) and security advisories.

Adopt CACAO playbooks. CACAO is a promising open standard. Subscribers should evaluate whether the CACAO standard eases maintaining the cybersecurity posture for their ICS. CACAO allows the definition of variables enabling a context-aware and asset-centric approach for quick and efficient remediation. For example, subscribers can define CACAO templates for action classes or even more specific operations, dynamically populate them with variables, and automatically generate context-aware playbooks. At the time of our research, security advisories are still premature, allowing only partial automation. However, implementing the CACAO standard is associated with great efforts. As long as there is no CACAO interpreter, subscribers must manually develop the CACAO playbook integration and execution. The main weak points of CACAO are the premature definitions, low adoption, and a small community.

Prioritize vulnerabilities. Organizations should think about prioritizing ICS vulnerabilities. In our small test environment, we faced several advisories and relevant CACAO playbooks leading to the questions of prioritization. Released in 2021, the Exploit Prediction Scoring System (EPSS) could be useful in ICS environments as it enables efficient prioritization of vulnerability remediation. EPSS considers exploit availability and likelihood, going beyond traditional vulnerability scores to identify critical vulnerabilities that require immediate attention. Given limited resources in ICS environments, EPSS allows operators to allocate resources more efficiently by focusing on vulnerabilities with higher exploitation likelihood, addressing the most critical risks first. Last, EPSS utilizes data from various sources, enhancing vulnerability prioritization accuracy, and providing reliable and actionable insights. In summary, EPSS empowers ICS operators to make informed vulnerability management decisions, safeguarding critical infrastructures effectively against potential cyber threats.

7 Conclusion

Security advisories for ICS vulnerabilities include alternative remediation measures when simply updating to the newest version is not an option. We have generated ICS vulnerability playbooks using open CSAF and CACAO standards. Our approach is the first to combine the fields of security advisories and playbooks addressing organizations' need to handle ICS vulnerabilities. While security advisories foster informing about vulnerabilities, playbooks are intended for workflow actions and eventually support automated execution. We have shown that crucial remediation advice can be included in CACAO playbooks by implementing a process model and experimenting with an industrial use case. ICS security advisories exist in various formats. Therefore, conversion to the CSAF standard is central to automated playbook generation. Toward the creation of individual workflow actions, we built upon matching terms to classify different remediation measures. In 79 security advisories, we identify a high prevalence of update advice and fewer practical remediation steps. Our results lead us to recommendations for security advisory publishers and automated vulnerability handling. Improving security advisories' structure and the content will help vulnerability playbook generation.

Future research can focus on further integration of open standards and their various features. While we use matching terms to extract workflow actions, artificial intelligence (e.g., large language models) might be able to build technical commands and add conditional workflow logic. Toward automated playbook execution, we also see the necessity to incorporate organization-specific factors as remediation measures could be deliberately kept vague to serve all architectures and systems equally well. Additionally, our work is based on available ICS data. As a result, our vulnerability playbooks are specific to ICS. It is worth investigating vulnerability playbook generation for IT assets. Future research could also compare more ICS advisories from plenty vendors to deepen the discussion on recommendations, but also measure the scalability of our vulnerability playbook generation process in larger environments. Another crucial aspect for future research is to target complex and dynamic documentation requirements in large scale ICS environments. Nevertheless, we see the two emerging open standards with increasing number of adopters shaping tomorrow's security operations.

Acknowledgements This research was supported by the German Federal Ministry of Education and Research (BMBF) as part of the DEVISE project and by the Bavarian Ministry of Economic Affairs, Regional Development and Energy (DIK0338/01) as part of the INSIST project. We thank Dehn SE for the great cooperation and feedback.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability The prototype and data supporting the findings of this research paper are openly available and accessible through a GitHub repository: https://github.com/ad2play/ad2play.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

A Industrial use case: Siemens device representation

This JSON-structured excerpt shows parts of the Siemens device comprising several hardware and software components, i.e., industrial automation systems SIMATIC or motion control systems SIMOTION. Each of these components has a dedicated CPE-ID for the software, hardware, and device tags in case the CPE-ID is unavailable.

```
{
    "thingId":"SOAR4IoT:Mock_Siemens",
    "policyId":"SOAR4IoT:policy",
    "attributes":{
        "manufacturerID":1,
        "manufacturerName":"Siemens",
        "dateCode":"",
        "type":"Mock",
        "security":{
            "cpe":[
                {
                    "device":"cpe:2.3:h:siemens:simatic_s7
                        -1200:-:*:*:*:*:*:*:*",
                    "firmware":"cpe:2.3:o:siemens:
                        simatic_s7_cpu_1200_firmware
                        :4.0:*:*:*:*:*:*:*"
                },
                {
                    "device":"cpe:2.3:a:siemens:simatic_s7
                        -1500:-:*:*:*:*:*:*:*",
                    "firmware":"cpe:2.3:a:siemens:simatic_s7-1500
                        __software_controller:-:*:*:*:*:*:*:*"
                },
                {
                    "device":"cpe:2.3:h:siemens:simatic_s7
                        -300:-:*:*:*:*:*:*:*",
                    "firmware":"cpe:2.3:o:siemens:
                        simatic_s7_300_cpu_firmware
                        :-:*:*:*:*:*:*:*"
                },
                {
                    "device":"cpe:2.3:h:siemens:simatic_s7
                        -400:-:*:*:*:*:*:*:*",
                    "firmware":"cpe:2.3:o:siemens:
                        simatic_s7_400_cpu_firmware
                        :-:*:*:*:*:*:*:*"
                },
                {
                    "device":"cpe:2.3:h:siemens:simatic_s7
                        -400:-:*:*:*:*:*:*:*",
                    "firmware":"cpe:2.3:o:siemens:
                        simatic_s7_400_cpu_firmware
                        :-:*:*:*:*:*:*:*"
                },
                {
                    "device":"",
                    "firmware":"cpe:2.3:a:siemens:simatic_step_7
                        :12.0:*:*:*:*:*:*:*"
                },
                {
                    "device":"",
                    "firmware":"cpe:2.3:a:siemens:simatic_s7-
                        plcsim_advanced:-:*:*:*:*:*:*:*"
                }
                [...]
            ],
            "entity_tags":[
                "Ad2Play:Mock_Siemens"
            ],
            "group_tags":[
                "Ad2Play:Twin_Group_1"
            ],
            "match_tags":[
                "SIMATIC S7-1200",
                "SIMATIC S7-1500",
                "SIMATIC S7-300",
                "SIMATIC S7-400",
                "STEP7 Professional",
                "STEP7 Safety Advanced",
                "SIMATIC STEP 7",
                "SIMATIC S7-PLCSIM Advanced",
                "SIMATIC Target for Simulink",
                "SIMATIC Safe Kinematics",
                "SIMATIC Kinematics Operate",
                "SINAMICS V20",
                "SINAMICS V90",
                "SINAMICS S210",
                [...]
            ]
        }
    }
}
```

B Excerpt of a generated CACAO Playbook

This CACAO playbook starts with the "Start Playbook" step and proceeds through various steps designed to handle specific actions related to a vulnerability. One of the steps is named "Access-Action-Step," which involves limiting access to Port 102/TCP to trusted users and systems only. The step includes several step variables, such as "action_description," "sentence_noun_tags," "entity_tags," and "group_tags," which provide specific information about the step. On successful completion of the "Access-Action-Step," it proceeds to another step.

```
{
        "_id": "62c44b364466fa24127ad4e7",
        "type": "playbook",
        "spec_version": "1.0",
        "id": "playbook—4e105ae9–e4b7–53e0–935a–
             fed1125ca376",
        "name": "AUTOGENERATED Playbook from sourced CSAF
             file(s) 62c3f0a499cf2533865814eb",
        "created_by": "identity—a9becb6a–d006–518a–a0a1–66
             a7bc70675e",
        "created": "2023–07–05T14:31:18.672Z",
        "external_references": [
            {
                "name": "CSAF File",
                "description": "Id of CSAF file that was
                     used for the creation of the playbook",
                "external_id": "62c3f0a499cf2533865814eb",
                "_id": "62c44b364466fa24127ad4e8"
            }
        ],
        "workflow_start": "step—e7d19860–a84a–563d–8705–02
             beb8f03441",
        "workflow": {
            "step—e7d19860–a84a–563d–8705–02beb8f03441": {
                "type": "start",
                "name": "Start Playbook",
                "on_completion": "step—3f0d51ad–7500–5dca
                     –90e2–8ed00a4d9a4f",
                "_id": "62c44b364466fa24127ad4e9",
            },
            [...]
            "step—a5f44526–2a7a–5dfb–820d–74ce00db5cf1": {
                "type": "single",
                "name": "Access–Action–Step",
                "step_variables": {
                    "[$$action_description$$]": {
                        "type": "string",
                        "description": "This is the sentence
                              that triggered the Pattern and
                              includes the action",
                        "value": "Limit access to Port 102/
                             TCP to trusted users and
                             systems only.",
                        "constant": true,
                        "_id": "62c44b364466fa24127ad4db"
                    },
                    "[$$sentence_noun_tags$$]": {
                        "type": "string",
                        "description": "This is the
                             stringified array of the nouns
                             used in the sentence that
                             triggered the pattern",
                        "value": "Limit,access,Port,TCP,
                             users,systems",
                        "constant": true,
                        "_id": "62c44b364466fa24127ad4dc"
                    },
                    "[$$entity_tags$$]": {
                        "type": "string",
                        "description": "These are the
                             entity_tags related to the
                             Twins",
                        "value": "Ad2Play:Mock_Siemens",
                        "constant": true,
                        "_id": "62c44b364466fa24127ad4dd"
                    },
                    "[$$group_tags$$]": {
                        "type": "string",
                        "description": "These are the
                             group_tags related to the Twins
                             ",
                        "value": "Ad2Play:Twin_Group_1",
                        "constant": true,
                        "_id": "62c44b364466fa24127ad4de"
                    }
                }
                [...]
            }
        }
    }
```

References

1. Lawson, C., Price, A.: market guide for security orchestration, automation and response solutions (2022)
2. Biden, J.R.J.: Executive order on improving the nation's cybersecurity. https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/. Last accessed 2023-06-05 (2022)
3. Forum of Incident Response and Security Teams (FIRST). Automation sig. https://www.first.org/global/sigs/automation/. Last accessed 2023-06-05 (2023)
4. Stevens, R., Votipka, D., Dykstra, J., Tomlinson, F., Quartararo, E., Ahern, C., Mazurek, M.L.: How ready is your ready? Assessing the usability of incident response playbook frameworks. In: Proceedings of the 2022 SIGCHI Conference on Human Factors in Computing Systems (CHI '22) (ACM), pp. 1–18. https://doi.org/10.1145/3491102.3517559 (2022)
5. OASIS. Cacao security playbooks version 1.0—committee specification 02. https://docs.oasis-open.org/cacao/security-playbooks/v1.0/security-playbooks-v1.0.html. Last accessed 2023-06-05 (2021)
6. Cybersecurity and Infrastructure Security Agency (CISA), Federal government cybersecurity incident and vulnerability response playbooks. Tech. rep., Cybersecurity and Infrastructure Security Agency (CISA) (2021)
7. Wang, B., Li, X., de Aguiar, L.P., Menasche, D.S., Shafiq, Z.: Characterizing and modeling patching practices of industrial control systems. In: Proceedings of the 2017 ACM on Measurement and Analysis of Computing Systems (POMACS '17), vol. 1(1), p. 1. https://doi.org/10.1145/3078505.3078524 (2017)
8. Cybersecurity & Infrastructure Security Agency (CISA). Ics-cert advisories. https://www.cisa.gov/uscert/ics/advisories. Last accessed 2023-06-05 (2023)
9. OASIS. Common security advisory framework version 2.0—committee specification 03. https://docs.oasis-open.org/csaf/csaf/v2.0/csaf-v2.0.html. Last accessed 2023-06-05 (2022)
10. National Vulnerability Database (NVD). Cve-2022-34819 detail. https://nvd.nist.gov/vuln/detail/CVE-2022-34819. Last accessed 2023-06-05 (2022)
11. Common Vulnerabilities and Exposures (CVE). https://cve.mitre.org/. Accessed 20 July 2023 (2023)
12. National Vulnerability Database (NVD) Common Platform Enumeration (CPE). https://nvd.nist.gov/products/cpe. Accessed 20 July 2023 (2023)

13. Common Vulnerability Scoring System (CVSS). https://www.first.org/cvss/. Accessed 20 July 2023 (2023)
14. OASIS Common Security Advisory Framework (CSAF) Technical Committee. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=csaf. Accessed 20 July 2023 (2023)
15. Skopik, F., Settanni, G., Fiedler, R.: A problem shared is a problem halved: a survey on the dimensions of collective cyber defense through security information sharing. Comput. Secur. **60**, 154 (2016). https://doi.org/10.1016/j.cose.2016.04.003
16. Gascon, H., Grobauer, B., Schreck, T., Rist, L., Arp, D., Rieck, K.: Mining attributed graphs for threat intelligence. In: Proceedings of the 7th ACM on Conference on Data and Application Security and Privacy (CODASPY '17) (ACM, 2017), pp. 15–22. https://doi.org/10.1145/3029806.3029811
17. Schlette, D., Caselli, M., Pernul, G.: A comparative study on cyber threat intelligence: the security incident response perspective. IEEE Commun. Surv. Tutor. **23**(4), 2525 (2021). https://doi.org/10.1109/COMST.2021.3117338
18. OASIS. Open command and control (OpenC2) language specification version 1.0—committee specification 02. https://docs.oasis-open.org/openc2/oc2ls/v1.0/oc2ls-v1.0.html. Last accessed 2023-06-05 (2019)
19. MITRE. Detection, denial, and disruption framework empowering network defense (D3FEND). https://d3fend.mitre.org/. Last accessed 2023-06-05 (2023)
20. ATC Project. RE&CT framework documentation. https://atc-project.github.io/atc-react/. Last accessed 2023-06-05 (2020)
21. West-Brown, M.J., Stikvoort, D., Kossakowski, K.P., Killcrece, G., Ruefle, R., Zajicek, M.: Handbook for computer security incident response teams (CSIRTs). Tech. rep., Defense Technical Information Center. https://doi.org/10.21236/ada413778 (2003)
22. Senthivel, S., Dhungana, S., Yoo, H., Ahmed, I., Roussev, V.: Denial of engineering operations attacks in industrial control systems. In: Proceedings of the 8th ACM Conference on Data and Application Security and Privacy (CODASPY '22) (ACM), CODASPY '18, pp. 319–329. https://doi.org/10.1145/3176258.3176319 (2018)
23. Ghena, B., Beyer, W., Hillaker, A., Pevarnek, J., Halderman, J.A.: Green lights forever: Analyzing the security of traffic infrastructure. In: Bratus, S., Lindner, F.F. (eds.), Proceedings of the 8th USENIX Workshop on Offensive Technologies (WOOT '14). USENIX Association (2014)
24. Li, F., Durumeric, Z., Czyz, J., Karami, M., Bailey, M., McCoy, D., Savage, S., Paxson, V.: You've got vulnerability: exploring effective vulnerability notifications. In: Holz, T., Savage, S. (eds.), Proceedings of the 25th USENIX Security Symposium (USENIX Security '16). USENIX Association, pp. 1033–1050 (2016)
25. Allodi, L., Banescu, S., Femmer, H., Beckers, K.: Identifying relevant information cues for vulnerability assessment using CVSS. In: Proceedings of the 8th ACM Conference on Data and Application Security and Privacy (CODASPY '18) (ACM), CODASPY '18, pp. 119–126 (2018). https://doi.org/10.1145/3176258.3176340
26. Fenz, S., Ekelhart, A., Weippl, E.: Semantic potential of existing security advisory standards. In: Proceedings of the FIRST 2008 Conference-Forum of Incident Response and Security Teams (FIRST '08) (2008)
27. Fenz, S., Ekelhart, A., Weippl, E.: Fortification of IT security by automatic security advisory processing. In: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications (AINA '08) (IEEE), pp. 575–582. https://doi.org/10.1109/aina.2008.69 (2008)
28. Challande, A., David, R., Renault, G.: Building a commit-level dataset of real-world vulnerabilities. In: Proceedings of the 12th ACM Conference on Data and Application Security and Privacy (CODASPY '22) (ACM), CODASPY '22, pp. 101–106. https://doi.org/10.1145/3508398.3511495 (2022)
29. Mavroeidis, V., Eis, P., Zadnik, M., Caselli, M., Jordan, B.: On the integration of course of action playbooks into shareable cyber threat intelligence. In: Proceedings of the 2021 IEEE International Conference on Big Data (Big Data '21) (IEEE), pp. 2104–2108. https://doi.org/10.1109/bigdata52589.2021.9671893 (2021)
30. Akbari Gurabi, M., 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 (ARES '22) (ACM), pp. 1–8. https://doi.org/10.1145/3538969.3544478 (2022)
31. Shaked, A., Cherdantseva, Y., Burnap, P.: Model-based incident response playbooks. In: Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES '22) (ACM), pp. 1–7. https://doi.org/10.1145/3538969.3538976 (2022)
32. Islam, C., Babar, M.A., Nepal, S.: A multi-vocal review of security orchestration. ACM Comput. Surv. **52**(2), 1 (2019). https://doi.org/10.1145/3305268
33. 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 (ARES '22) (ACM), pp. 1–10. https://doi.org/10.1145/3538969.3538975 (2022)
34. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. J. Manag. Inf. Syst. **24**(3), 45 (2007). https://doi.org/10.2753/MIS0742-1222240302
35. Dietz, M., Pernul, G.: Digital twin: empowering enterprises towards a system-of-systems approach. Bus. Inf. Syst. Eng. **62**(2), 179 (2020). https://doi.org/10.1007/s12599-019-00624-0
36. Schlette, D., Menges, F., Baumer, T., Pernul, G.: Security enumerations for cyber-physical systems. In: Singhal, A., Vaidya, J. (eds.), Data and Applications Security and Privacy XXXIV—34th Annual IFIP WG 11.3 Conference, DBSec: Regensburg, Germany, June 25–26, 2020, Proceedings, Lecture Notes in Computer Science, vol. 12122, pp. 64–76. Springer (2020). https://doi.org/10.1007/978-3-030-49669-2
37. National vulnerability database. https://nvd.nist.gov/. Accessed on 20 Jul 2023

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.