AdvisoryHub: Design and Evaluation of a Cross-Platform Security Advisory System for Cyber Situational Awareness

Marc-André Kaufhold(✉), Julian Bäumler, Nicolai Koukal, and Christian Reuter(✉)

Science and Technology for Peace and Security (PEASEC), Technical University of Darmstadt, Pankratiusstraße 2, 64289 Darmstadt, Germany
{kaufhold,baeumler,koukal,reuter}@peasec.tu-darmstadt.de

Abstract. Computer Emergency Response Teams (CERTs) provide advisory, preventive, and reactive cybersecurity services for authorities, citizens, and businesses. However, their responsibility of establishing cyber situational awareness by monitoring and analyzing security advisories and vulnerabilities has become challenging due to the growing volume of information disseminated through public channels. Thus, this paper analyzes semi-structured interviews (N = 17) with CERT employees to identify user requirements, which are translated into the design of a system for automatically retrieving and extracting security advisory documents from Common Security Advisory Framework (CSAF), HTML, and RSS sources. The evaluation using a CERT-based list of trusted security advisory sources (N = 53) shows that the developed system can retrieve 90% of the published advisory documents, which is a significant improvement over systems only relying on the retrieval from RSS feeds (30%).

Keywords: Cyber Situational Awareness · Security Advisories · Computer Emergency Response Teams

1 Introduction

The importance of cybersecurity is not only motivated by the advancing digitization and networking of society but also by the increasing frequency and sophistication of cyberattacks [8]. Recognizing the need for incident management, Computer Emergency Response Teams (CERTs) have been established in public and private sectors [20] to provide reactive, proactive, and security quality services internally or externally for authorities, citizens, and enterprises [10]. To provide these services, CERTs must first establish Cyber Situational Awareness (CSA) by monitoring, analyzing, and communicating cyber threats and security vulnerabilities [6]. CSA describes a level of understanding possessed by individuals that allows them to perceive pertinent elements in the cyber environment within a defined timeframe and spatial context, interpret their significance, and anticipate their future status [7].

M. Dalla Preda et al. (Eds.): ARES 2025, LNCS 15993, pp. 47–68, 2025.
https://doi.org/10.1007/978-3-032-00627-1_3

Yet, the establishment of CSA is becoming more difficult due to the increasing volume of information accessible through public channels, including feeds, social media, vulnerability databases, third-party services, and websites [8]. To enhance CSA, security advisories provide vulnerability information, enabling users to identify vulnerable products and services and take action to remediate the vulnerabilities [13]. Empirical studies with German state CERTs indicate a lack of efficient mechanisms for extracting and seamlessly incorporating real-time threat intelligence [20]. Moreover, CERTs often encounter irrelevant, duplicated, and occasionally implausible information, whose curation for stakeholder reports can consume multiple hours per day [9].

However, we identified a lack of design studies focusing on tools for the cross-platform collection and analysis of security advisories [8]. Thus, this paper examines the following research question: **What are user requirements and design implications for a cross-platform security advisory content retrieval and extraction system to facilitate the cyber situational awareness of CERTs?** Based on a literature review (Sect. 2) and semi-structured interviews (N = 17) with CERT employees (Sect. 3), this paper analyses the context of use and specifies user requirements (Sect. 4) to describe the design of a cross-platform security advisory system (Sect. 5). Thereafter, it presents the evaluation with security advisory sources (N = 53), highlighting the successful retrieval of 90% of the published advisory documents (Sect. 6), and discusses the findings of our first design iteration (Sect. 7).

2 Related Work

CERTs need to overview the general threat landscape to align organizational security measures and provide services to external clients, requiring the identification of suspicious behaviours, current information on external threats, and participation in security communities to stay updated on emerging threats [14].

2.1 Distribution and Standardization of Security Advisories

Thus, numerous manufacturers, public sector CERTs and other private sector organizations release information regarding vulnerabilities and solutions, i.e. security advisories, through their own communication channels [13]. These platforms can be divided into sources, aggregators, and hybrid sources/aggregators of information [16]. Product CERTs primarily publish advisories which are focused on concrete vulnerabilities related to their products and are thus classified as sources of advisories. In contrast, publications from public sector CERTs typically reference advisories first published by the product CERTs. Acting as advisory aggregators, they often group related vulnerabilities into one report, e.g., when the same type of vulnerability was found in multiple products within a narrow time frame. However, existing studies outline the lack of standardization regarding the representation and provision of security advisories, forcing German public CERTs to gather security advisories from diverse manufacturer websites manually on a daily basis [20], as obstacles for deploying security-relevant updates in a timely manner. Although some vendors have been providing security advisories via email subscriptions or RSS feeds, a considerable amount of daily manual checking for new security advisories is still needed in order to prepare stakeholder reports or warnings. Thus, the Common Security Advisory Framework (CSAF) was developed recently and is supported by other well-known organizations such as Redhat, Cisco, and the German Federal Office for Information Security (BSI) [23]. It is based on JSON and supports the use of CPE, CVSS, and CWE. Vulnerabilities may be referenced using the CVE system, or other forms of vulnerability IDs, such as vendor-specific systems, may be used. Additionally, the Traffic Light Protocol (TLP) can be used to label advisory information regarding their sensitivity to sharing.

2.2 Crawling and Processing of Security Advisories

Due to the relative novelty and thus lacking distribution of the CSAF standard, multiple approaches have been designed to gather security advisories differently. [5] proposed a system that collects security advisories, converts them into a standardized and machine-readable format, and distributes them to its subscribers. Compared to text-based security advisory systems, the system offers precise automated filtering and enables the user to map the advisories to the existing infrastructure. The authors stated that the message transformation module should contain conversion rules that convert the unstructured web data into structured vulnerability data for each data source. The system, however, does not automatically create these extraction rules, which hinders the scalability of advisory sources and only supports the retrieval of advisories via RSS. [9] have developed an automatic real-time cybersecurity dashboard that ingests, processes, and displays not only security advisory data, but also indicators of compromise, social media data, and data from vulnerability databases. Furthermore, Taranis3 was developed to assist in retrieving, structuring, and analyzing vulnerability information from various sources as well as writing and publishing advisories [19]. It supports the retrieval of web, email, and social media-based sources. While the retrieved documents can be automatically clustered, the content of security advisory documents is not parsed. As a consequence, the analysis of security advisories is only semi-automated, as the analysis process still requires significant manual effort. While SK-CERT created the Python-based tool Taranis NG, the latest iteration of the tool is Taranis AI, which aims to integrate modern natural language processing (NLP) methods into the analysis process, allowing the efficient identification and extraction of relevant entities [21].

3 Problem Identification: Empirical Pre-Study

To better understand how CERTs use security advisory documents and how this process could be enhanced by technology, transcripts of semi-structured interviews ($N = 17$) were analyzed. These were conducted prior to this work and focused not only on security advisory documents but, more broadly, on the technology use and processes in CERT operations. While this prior work [20] outlines collaborative, organizational and technological challenges and implications concerning German CERTs, it does not derive user requirements for and the actual design of a security advisory system. The interview guidelines encompassed (1) the roles and affiliations of the interviewees, (2) their procedures for reporting cyber incidents, (3) methods for monitoring cyber incident data, such as indicators of compromise, (4) processes for analyzing, prioritizing, and validating collected evidence, (5) collaborations among CERTs, and (6) how recommendations and warnings are communicated. We used a purposive sampling strategy in order to primarily involve personnel on operational (e.g., incident managers as technology operators), but also tactical level (e.g., internal team leaders) among German state CERTs. In summary, our interviews comprised eight internal incident managers (I01, I04-I07, I14, I17), six internal team leaders (I02, I08-I10, I12, I16), three external information security officers (I11, I12, I15), and one external public safety answering point (I13). Overall, participants (15 male and two female) from eleven organizations, including eight CERTs and three other organizations, were included in our pre-study interviews. Each interview session, conducted with the acceptance and informed consent of the participants, lasted approximately 60 min.

3.1 Data Analysis

The analysis was conducted by three researchers (two white males and one white female) from the domains of HCI, CSCW, and information security. Given its flexibility, *thematic analysis* appeared to be a well-suited approach for understanding how CERT staff work with security advisory documents [2]. The themes that form the core of thematic analysis can be identified either inductively or deductively. Given that there is a theoretical framework for establishing situational awareness but no framework for processing advisory documents, the use of inductive theme discovery seemed more appropriate. We followed the step model of inductive category development by [15], which comprises the determination of category definition, formulation of inductive categories, revision of categories, and a final working through the texts before interpreting and presenting the results. We clustered the themes into meta themes to present the themes more clearly and to show content-related references between the identified themes that were lost during the categorization process (Table 1). Although the original thematic analysis by [2] does not include such a procedure, [15] allows the use of main categories, which reflects our idea of meta themes. During biweekly team meetings, we discussed and revised these codes until we reached consensus.

3.2 Results

Both the **selection of relevant information sources and the retrieval of the vulnerability information** are challenging for CERT staff. For example,

Table 1. Identified themes, meta themes, and prevalences

Themes	Meta Themes	n
Sources, Social Media, Privacy	Sources Selection & Advisory Retrieval	57
Dissemination	Reporting & Spread of Information	40
Classification, Cognitive Aspects	Vulnerability Scoring & Classification	41
Time, Interfaces, Tools	(Lack of) Automation	44
Assets	IT Asset Management (ITAM)	20
Timeliness, Quality, Deduplication	Information Quality	27
Context, Summarization, Trends	Manual Vulnerability Analysis	15

one CERT employee stated that the lack of a catalogue of relevant sources is perceived as a problem: “[...] you would simply have to put together a catalogue of sites that you want to monitor. This certainly also happens in a different context, when in the business sector you want to be aware of certain things that someone publishes and that is difficult” (I10). According to six participants, their CERTs mostly rely on self-curated lists of sources and use information provided by the German national CERT (CERT-Bund) or other upstream CERTs. Some CERTs also use information provided by other CERTs, such as by the US-CERT, to close potential blind spots in the data. In addition to information provided by other CERTs, five employees report the use of information provided by hard- and software vendors. This information is usually retrieved manually from the vendor’s websites. One respondent stated that they rely solely on vendors actively alerting them of new vulnerabilities according to contractual obligations. Other sources of information include RSS feeds, mailing lists, online IT news, and social media. Three interviewees specifically stated that they try to use as much information as possible to maximize their situational awareness.

According to seven participants, nearly all CERTs **disseminate situational reports** to their stakeholders. Most commonly, a daily report is generated. Reports of lower frequency are sometimes created additionally or alternatively to daily reports: “Yes, we have a wide range of recommendations that we send out, so we create various products for our customers. These are, on the one hand, a situational report - that is a vulnerability report - and, more recently, a cyber situational report where we try [...] to provide information that is tailored to the target group and clientele at various levels” (I01). The reports contain vulnerability details, information about incidents that affected the network, and information about threats that are not directly related to vulnerabilities, e.g., in the case of recent phishing waves. The reports often contain a high-level threat indicator, such as a traffic light: “Yes, [the situation report includes] the incident reports, vulnerabilities, and exploits. General information from the internet, e.g., what is being discussed in the press, what security topics can be found in the news, and of course, the general topic of IT security, what is a current topic in the press that is being discussed, should be included there” (I05). Three participants stressed that important elements of the reports are the CERT’s recommendations or instructions on what steps should be taken in response to the vulnerability, which help to achieve the overall goal of remediating the vulnerabilities as soon as possible. Some CERTs contact the operational teams directly in cases of severe vulnerabilities, e.g., via email or phone, so that the vulnerabilities can be mitigated before the next situational report is sent.

CERTs **score, categorize, and classify vulnerabilities** in terms of their severity, relevance, and potential impact on handling cognitive challenges of CSA, such as information overload and alert fatigue. One example of such a score is the CVSS. According to five participants, the initial score provided by the vendors or the upstream CERT is an important input for the scoring process conducted by the CERT. However, due to differences in infrastructure in different organizations, there is a subjective component in scoring systems. For that reason, some CERTs incorporate their own logic and weights into the scoring process, such as disregarding vulnerabilities for products which are not used by their constituents: "[...] so of course, we only try to subscribe to information that is relevant to us or to our target groups, as we usually call them. So we have a very good overview of the software that is used in the state, of course not in detail" (I10). Some CERTs use information on how often a product is used in the organization or information on the patch level of the infrastructure. Sometimes, it is also considered whether a vulnerability is actively being exploited in the wild. Four respondents reported that the calculated scores and classifications are used to sort, filter, and prioritize vulnerabilities.

CERTs are interested in a **higher degree of automation** in vulnerability information retrieval and processing, which is discussed by six interviewees. The first reason is that manual labour can be saved to improve in-depth analyses of selected vulnerabilities and achieve a better overall analysis result. If vulnerability information retrieval processes are not automated, they may take between 0.5h and 1.5h per day. Smaller organizations, in particular, cannot devote that many resources to retrieving vulnerability information. The second reason is that CERTs could shorten query intervals. With manual retrieval, sources are typically queried and inspected once a day. In addition to retrieval, six participants would like to use automation to merge information from different sources, determine the reliability of sources, simplify report creation, fill in missing information in vulnerability descriptions, or remove redundant information: "It would be a huge relief if the information from different sources were brought together [automatically] in one place and an evaluation could somehow be created from this" (I15). There have been some concerns about information accuracy, particularly in cases where machine learning technologies are used for the prioritization of vulnerabilities. Two respondents thus suggested displaying percentage-based confidence levels and data sources of the algorithm alongside the information generated. According to four interviewees, the most difficult challenge for automation is that security advisories are not offered in a standardized format. Furthermore, due to a lack of standardization, the data format in which a vendor publishes their advisories in may change at any time. This would then break possible automation solutions.

Furthermore, it is essential for CERTs to **know whether a particular product (both software and hardware) is being used** by the constituents to reliably predict the risk of new vulnerabilities. As mentioned by five participants, CERTs often do not have a complete picture of the products their constituents use: “So it’s just way too much that I really know what software is used here. I’m aware of maybe 80 per cent, maybe 90, but I can’t reach 100” (I11). Some CERTs warn not only about vulnerabilities affecting software that runs on organization-managed devices but also software that is likely to run on the employees’ personal devices. IT asset management is important for both knowing whether a certain product is used within an organization, and also for identifying the responsible technical contact that is necessary for mitigating vulnerabilities and other incidents. This information can be used to send vulnerability information only to those who are affected by them and thereby ensure that the report is actionable for the receiver: “So if we were, for example, informed of a critical SAP error, then we would only send it to [department name] because the [department name] has an SAP [instance] running and we know that no one else uses it” (I11). Some CERTs stated that they have minimal information about what software is used by their constituents, and they thus forward all vulnerabilities without changing the scores provided by the vendors or upstream CERTs.

Various aspects of **information quality** were mentioned during the interviews, especially timeliness. For example, three CERT employees specifically mentioned that in addition to the information provided by the national CERT, vendor websites are monitored to avoid the national CERT’s reporting delay: “Usually, we set the criticality higher and next we check around a hundred websites for security advisories so that we can simply publish security warnings before the BSI [publishes the vulnerabilities]” (I02). The timeliness of the information enables CERTs to remediate the vulnerabilities as quickly as possible. Besides timeliness, it is also important that the processed information is dated properly. This allows for displaying the course of events chronologically and helps in the selection process when there are two conflicting pieces of information. Another important quality metric is the correctness of information. Since the accuracy often cannot easily be verified by the CERTs, they deploy different strategies to determine the factuality of information. One CERT relies primarily on vendor advisories as an authoritative source of information: “Of course, we also look through the normal standard sources that every other user uses. Sometimes something is discovered that isn’t yet published on the vendor’s website or additional information. Those are the things [we additionally use], but they are all things that we then confirm through the vendors” (I13). More generally, five participants discussed the importance of determining the primary source in order to gain insight into the reliability of information. Using and cross-checking multiple sources can help with improving the reliability of information. In cases where automation is used, it is not only important to know where the underlying information came from, but also how a system derived a result from that data. Some CERTs also use social media as an indicator that some information may be more reliable than others. One concern raised regarding the use of machine learning for automation was the risk of biases. As stressed by five participants, redundant and duplicated data is a common issue related to data quality in CERT processes. The interviewees suggested different solutions to this problem, such as more standardization or the use of artificial intelligence.

There are various steps which CERTs take to gain CSA, including the often **manual vulnerability analysis** and the preparation of the information for dissemination. According to three participants, CERTs add recommendations to security advisories to ensure that their constituents, the receivers of the disseminated reports, can work with actionable information. In cases where CERTs use information from their upstream CERTs, they supplement and enrich the information to better match the needs of their constituents. Similarly, sometimes the information is summarized and unnecessary information is removed. They also may merge information from different sources and reference the respective original publications in their report: “The vulnerabilities of vendors are copied, briefly summarized and tagged with a link so that further information can be looked up again by the interested party who receives it” (I14).

4 Defining the Objectives of Solution: Conceptualization

Based on the literature review and interview data, we envisioned a potential pipeline for the technical processing of security advisories (Fig. 1) that is based on functional requirements (R1-10), which can also be mapped to non-functional requirements (U1-11). To identify both functional and non-functional requirements, we first analyzed the empirical material alongside the identified themes and then correlated it with selected related work, which is detailed in our GitHub appendix [12]. The reason for this approach was that the literature mainly dealt with threat-focused but not vulnerability-focused CSA and could, therefore, only be supplementary because many literature findings did not apply to the topic of investigation.

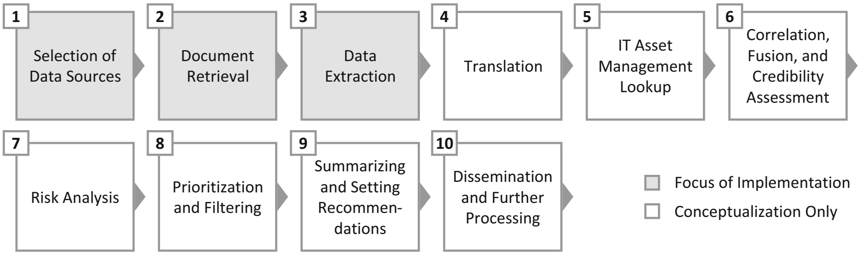

1
Selection of Data Sources
2
Document Retrieval
3
Data Extraction
4
Translation
5
IT Asset Management Lookup
6
Correlation, Fusion, and Credibility Assessment
7
Risk Analysis
8
Prioritization and Filtering
9
Summarizing and Setting Recommendations
10
Dissemination and Further Processing
Focus of Implementation
Conceptualization Only

Fig. 1. Steps for handling security advisory documents.

4.1 Functional Requirements

First, there should be a comprehensive prefilled catalogue of relevant sources (R1, Facilitate the Selection of Security Advisory Sources), which should be monitored for automated advisory retrieval (R2, Continuously Monitor Security Advisory Sources for Document Retrieval) and then transformed into a standardized format (R3, Extract and Convert Data into a Standardized Format). Considering that advisories can originate from different national CERTs, language translation steps using specific APIs or code libraries (R4, Translate Foreign Language Content for the System Operator) might be required before automatically determining whether a vulnerability is relevant for the respective stakeholder group (R5, Integrate an IT Asset Management to Check Stakeholder Affectedness), or even merging complementary data from different advisory sources might be advisable (R6, Enable the Fusion of Data from Multiple Sources). In case of a relevant security vulnerability, there is a need to assess the criticality, timeliness, and risk of compromise (R7, Support the Risk Analysis of Vulnerability Exploitation) before prioritizing the urgency of concurrent threats (R8, Provide Functionality for Sorting, Prioritizing and Filtering Security Advisories). Given the potentially large amount of security advisories, a functionality for recommendations and summarization (R9, Summarize Security Advisory Content and Provide Recommendations for Action) and support for creating and disseminating reports (R10, Facilitate the Creation and Dissemination of Stakeholder Reports) was requested. In Sect. 5, we present the first design iteration of a system supporting the envisioned pipeline's first three steps.

4.2 Non-functional Requirements

Besides the functional requirements that describe *which* steps need to be performed by the system, there are non-functional requirements that describe *how* the steps must be carried out. The non-functional requirements include ensuring data quality (U1) and maintaining data origin (U2), which involves the trustworthiness, completeness, and freshness of data. Trustworthiness can be enhanced by correlating data from multiple sources and maintaining data origin information. Automation (U3) and resilience (U4) are essential for efficient data collection. The system must handle incomplete or contradictory data by calculating missing data or making predictions. Timeliness (U5) and timestampedness (U6) are critical, as faster retrieval and interpretation of vulnerability information enable more frequent querying. Precise timestamps are necessary for creating accurate event timelines. Standardization (U7) and configurability (U8) are required for consistent communication between pipeline steps, with the core data format aligning with CSAF to ensure all information is conveyed effectively. Scalability (U9) is important as the architecture must support a high data volume from diverse sources. Finally, the system should be adaptable to the cognitive needs of users (U10), presenting only necessary information to avoid overload.

4.3 Mapping of Pipeline Steps to Non-functional Requirements

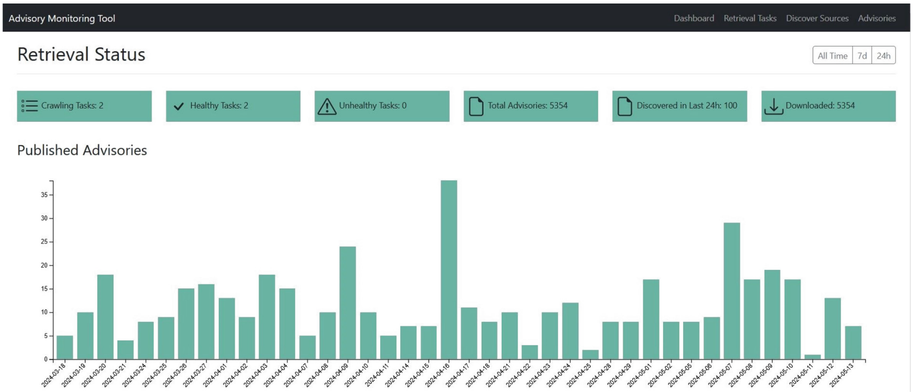
The identified non-functional requirements have an impact on the functional requirements and, therefore, also on the pipeline steps. However, not all non-functional requirements are applicable to every step. To visualize dependencies between non-functional requirements and pipeline steps, an approach similar to Quality Function Deployment (QFD) is chosen, which is a method for mapping customer requirements to engineering requirements [17]. We opted for this approach because it helps to clearly separate functional and non-functional requirements (which were mentioned intermixed, especially in the interviews) along the two axes of the matrix (Table 2). It shows which non-functional requirements are related to which functional requirements (e.g., that the non-functional requirement of adaptability to the user's cognitive needs is not related to the front steps of the pipeline), which should help in the later implementation. It is also helpful to show relationships between the individual functional requirements and thus define a sequence of pipeline steps (e.g., the risk calculation step requires standardized data, so the earlier step of data extraction must establish standardization). The mapping of pipeline steps to non-functional requirements in Table 2 visualized dependencies and technical implications, showing how different steps enable (e), require (r), or are challenged (c) by these requirements.

Table 2. Mapping pipeline steps to non-functional requirements, showing how different steps enable (e), require (r), or are challenged (c) by these requirements.

⟶	Source Selection	Document Retrieval	Data Extraction	Translation	Asset Management	Correlation Fusion	Risk Calculation	Prioritization	Summarizing	Dissemination
Data Quality			e			e	r		c	r
Data Origin Principle		e	e	e		e				
Automation	e	e	e	e	e	e	e	e	c	e
Resilience			c		c	e				
Timeliness		e								
Timestampedness		e	e				r			
Standardization			e		r	r	r			r
Configurability	e	e	e	e	e	e	e	e	e	e
Scaleability	e	e				r				
Cognitive Adaptability								e	e	

5 Design: Cross-Platform Security Advisory System

Overall, we designed a Python application whose code and implementation details are published at GitHub [12]. The frontend web interface for configuring monitoring and retrieval of the advisory sources was implemented using the Bootstrap framework in combination with the JavaScript library D3.js, which was utilized to render charts. The backend is divided into four layers. The app layer forms the counterpart to the frontend by receiving the HTTP requests and responding to them accordingly. The batch layer is responsible for continuously running processes in the background to monitor **overview pages** (e.g., lists or tables) that contain references to advisory document detail pages, as well as monitoring these **detail pages** (including detailed attributes) and analyzing the retrieved content. To retrieve overview pages and process the content on the detail pages, the monitoring functions of the batch layer spawn retriever, extractor, and constructor function in the parallel layer. The data layer is responsible for persisting both the configuration data and the retrieved security advisory documents. While the configuration data and selected metadata are stored in an SQLite database, the retrieved security advisory documents are stored as files on disk. While the overall system is designed to address the functional requirements of data source selection (R1), document retrieval (R2) and data extraction (R3), we will discuss the consideration of non-functional requirements within the following subsections.



Retrieval Status

Crawling Tasks: 2

Healthy Tasks: 2

Unhealthy Tasks: 0

Total Advisories: 5354

Discovered in Last 24h: 100

Downloaded: 5354

Published Advisories

Fig. 2. Screenshot of the dashboard view.

5.1 Backend: Monitoring of Overview Pages

The system is able to monitor a variety of different data sources such as CSAF, HTML, and RSS content. To account for the differences between the data source categories, different retriever classes were developed. Each of these retriever classes ultimately inherits from the Python class *threading.thread* to enable asynchronous and parallel retrieval. First, the **HTML Table Retriever** allows the system to extract data from static websites containing the desired data within HTML table tags. The rows of these tables represent the advisory detail pages and their columns represent advisory metadata, including a link to the respective detail page for the advisory document. Second, the **HTML XPATH Retriever** allows the system to create detail tasks not only from table tags but arbitrary HTML elements that are part of the Document Object Model (DOM) of the monitored site. A representative application for this retriever are sites that contain multiple div elements that represent the advisories and their respective metadata attributes. The elements of interest are selected via the XPATH expression language. For example, the XPATH string *//div[contains(@class, 'advisory')]* can be used to select all div elements in the page DOM that have the string advisory in their class name and subsequently extract relevant values. Third, the **HTML URL Retriever** acts similarly to the XPATH retriever but only extracts a-tags from the DOM. To filter out references to pages that are not detail pages, the results are filtered with a prefix provided by the user. This prefix must be common to all the detail pages; however, it must be specific enough to distinguish detail pages from not detail pages. An example could be a string such as *http://www.example.com/advisories/*. The URL retriever is more robust than the Table retriever, but it is not able to extract the same amount of metadata provided by the detail page, such as the publishing data or the affected products.

Fourth, the **HTML Ajax Retriever** allows the system to extract the advisory information directly from JSON APIs that are called by the JavaScript code to populate the overview pages. For setting up the retriever, the user must manually inspect the overview site and determine the URL and the request parameters of the called API. This information is provided to the retriever to request the JSON data containing the desired information. The retriever is able to automatically find the required data within the JSON response and transform it into a table. After transforming relative URLs into absolute URLs, the selected table can be processed in the same way the other HTML retriever classes process the constructed table. Fifth, the **RSS Retriever** transforms an XML-based RSS feed into a list of elements representing the different advisories. The list is then transformed into a table and used to construct detail tasks, as in the other tables. The advisory content, which is often already part of the RSS feed, is stored in the detail task content field. Finally, the **CSAF Retriever** transforms the JSON-based CSAF feed into a table, which is used to construct detail tasks. Overall, the designed backend seeks to ensure data quality (U1) by trying the best possible retrievers for the construction of as complete as possible, standardized (U7) CSAF files, while maintaining the data origin principle (U2) by referencing the URL from which it was retrieved. Furthermore, it lays the foundation for automatic (U3) cross-platform retrieval of security advisories and also provides resilience (U4) against changes because the system can at least fall back to less optimal retrievers.

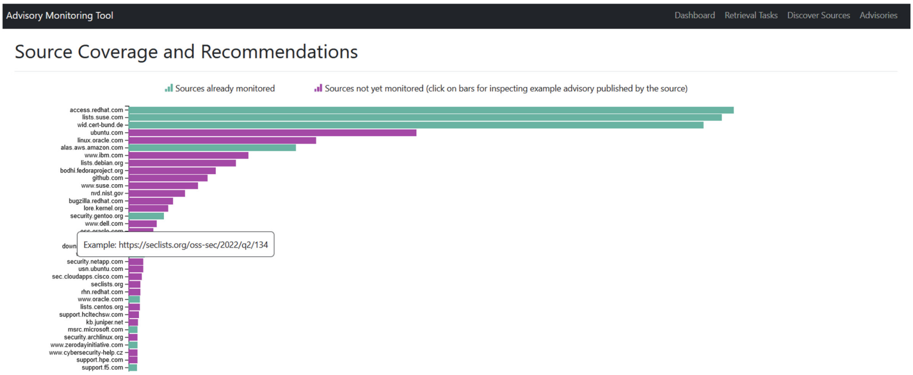


Fig. 3. Suggestions for additional sources to monitor.

5.2 Frontend: Cross-Platform Management of Security Advisories

The interface starts with a **dashboard page**, which consists of status elements indicating whether the retriever works properly and metrics, such as the number of new advisory documents discovered in the last 24 h (Fig. 2). Additionally, there is a time-series column chart showing how many advisory documents were published on a given date. This allows the user to quickly identify abnormalities, such as a higher-than-usual number of new vulnerabilities. By clicking on the date in question, the user can investigate the advisories published on that date in the preview menu.

On the **overview task page**, the user can inspect the running tasks. By clicking on the individual tasks, the user can inspect the task properties, such as the configured request body and header or error messages in case the retriever failed to poll a data source. New sources can be either added individually via a web form or in bulk via a CSV import function. When adding new sources via the web form, the user must first select the retriever type the system should use and then set the other task properties. After submitting the task, a test request is conducted to verify that the task was configured properly. If the task was configured incorrectly, or the polled server returned for other reasons, the user is displayed an error message. In case the request worked, the user needs to set column types in the tabularized form of the response. The system automatically suggests column types based on the column names coming from the server. After the column types, e.g., *URL*, *title*, or *identifier* were set, the user can add the task to the database. The system then automatically executes the retrieval in the background and creates detailed tasks if new advisory documents are detected.

The system automatically suggests new data sources based on the references of the documents already ingested and analyzed. In the **recommendation view**, which can be seen in Fig. 3, the user sees how often a particular provider was referenced by the ingested advisories. If there is already a monitoring task for that vendor, the respective bar is marked as green; otherwise it is marked red. The user can, therefore, screen the suggested sources by clicking on the red bars. This forwards the user to one example referenced detail page. After deciding whether the advisory source may be of interest, the user needs to navigate to the overview site of the provider and add it as a new task.

The implementation also includes a **preview page** that displays the retrieved and extracted advisory documents. The user can access the original resource by clicking on the link symbol left of the title. By clicking on the provider of an advisory document, the user can filter for documents that were published by that particular provider. This functionality is also offered for the name of the affected vendor and CVE numbers. The preview does not contain all advisory attributes, but they are stored in the CSAF files that were saved to disk. Overall, the designed frontend enhances the automation (U3) of security advisory gathering and ensures timeliness (U4) since new security advisories are constantly added to the dashboard. Furthermore, it facilitates the configuration (U7) of plenty of data sources, which is supplemented by the recommendations view, and thus constitutes a scalable (U9) cross-platform system for security advisory retrieval.

6 Demonstration: Formative Lab Evaluation

The evaluation aims to prove the utility and efficacy of the designed and implemented artefact. For choosing an appropriate evaluation methodology, [22] distinguish between naturalistic and artificial contexts, such as with real users of the artefact or in a simulated environment, as well as whether an evaluation is conducted ex-ante or ex-post, i.e. before or after the artefact is created. While an artificial evaluation comes with a lower cost compared to a naturalistic evaluation, it is not as well-suited to evaluate sociotechnical artefacts. Since the development of the system is in its first of multiple design iterations, the selected evaluation methodology should also result in knowledge that can be used formatively to improve future iterations of the artifact. To ensure the technical functionality of the system, an artificial evaluation seems more appropriate for the first round of evaluation. Given that the evaluation is based on the developed artefact, an ex-post evaluation can be conducted. First, the ability of the system to monitor sources and, second, the ability of the system to extract information from retrieved documents is evaluated by the third author.

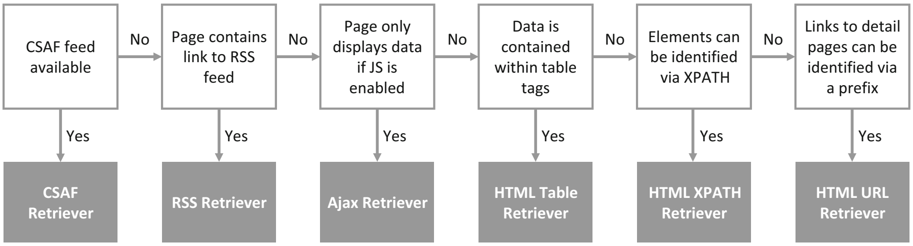

CSAF feed available
No
Page contains link to RSS feed
No
Page only displays data if JS is enabled
No
Data is contained within table tags
No
Elements can be identified via XPATH
No
Links to detail pages can be identified via a prefix
Yes
Yes
Yes
Yes
Yes
Yes
CSAF Retriever
RSS Retriever
Ajax Retriever
HTML Table Retriever
HTML XPATH Retriever
HTML URL Retriever

Fig. 4. Application of the retriever classes during the evaluation.

6.1 Retrieval Functionality Evaluation

The retrieval functionality is evaluated in a lab setting modelled after real challenges a CERT may face when working with advisory documents. The lab experiment builds upon a list of links to advisory providers that was made available by one of the CERTs involved in the interviews. For the evaluation, every advisory provider on the list is checked for retrievability and whether it can be monitored by the developed system. A provider is considered to be retrieved successfully if at least one of the retrievers can successfully be configured to generate detail tasks based on the page content. For selecting the appropriate retrievers, the decision scheme depicted in Fig. 4 was developed. The order of the decision scheme is based on the individual advantages and disadvantages of the retrievers, which are illustrated in Table 5. For example, as the CSAF retriever provides the most comprehensive data and the greatest stability over time against changes in templates, the user should first check if a CSAF feed is offered by the provider.

Table 3. Ability of retrievers to monitor overview pages

n = 53	CSAF	RSS	AJAX	TABLE	XPATH	URL	FAIL
Fetched	4	16	9	10	19	26	-
Best	4	13	6	8	9	8	5

The list provided by the CERT contained 53 links pointing primarily to pages on which security advisory information is published, but also pages on which new software releases are announced. The results documented in Table 3 show that the CSAF feeds are still not a common way of publishing security advisory information, as only 4 of the 53 surveyed sources offered advisories in the CSAF format. Of those sources, two were services provided by national CERTs and the other two were product CERTs publishing advisories. The results also indicate that approximately 30% of the surveyed data sources offer the option to subscribe to an RSS feed, which in consequence also means that cyber situational awareness systems only ingesting RSS data can merely retrieve 30% of the published vulnerabilities. The developed system, on the other hand, was able to monitor 90% of the advisory sources, which is a significant improvement. The 10% that could not be monitored consisted of sites that did not follow the overview-page-detail-page structure but instead contained all vulnerability data within a single page. Also, there were some JavaScript-dependent websites, where the API data could not be accessed. For example, one site executed JavaScript code to generate a token that was required for the API request.

6.2 Extraction and Standardization Evaluation

The evaluation of the extraction and standardization functionality was conducted similarly to the evaluation of the retrieval function. For each of the 48 sites that were successfully monitored by the retrieval modules, one random extracted document was manually inspected and rated regarding the quality of the extraction of its most important attributes. The attributes selected were title, vendor, product, text content, a timeline consisting of publication and update dates, a severity string, a CVSS score, and references. The extraction quality was rated on a scale from 0 to 3, with 0 meaning an attribute was not extracted or a wrong value was extracted that would be detrimental to the situational awareness of the operator. The used coding is detailed in our GitHub appendix [12]. One example of such a case would be the extraction of a CVSS score of 2.1 when the real score was 9.3 since that could lead to the advisory document being omitted by an automatic filtering system or being disregarded by a human operator. While value 1 was used for extractions that contained some correct pieces of information but were mainly incorrect, value 2 was used for extracted attributes that were inaccurate but were not detrimental to establishing situational awareness. An example of this would be the extraction of an unrelated link as a reference. The value 3 was used for cases where the extracted value was identical to the value a human would extract.

Table 4. Extraction Performance for Advisory Attributes

n = 48	NA	#0	#1	#2	#3	AVG
Vendor	1	8	0	10	29	2,28
Product	1	23	0	5	19	1,43
Vulnerability	8	9	0	1	30	2,30
Timeline	0	16	3	6	23	1,75
Title	0	4	1	2	41	2,67
Content	3	7	5	8	25	2,13
Severity	14	10	2	4	18	1,88
Score	25	9	0	1	13	1,78
References	3	12	5	9	19	1,78

Table 4 shows that the extraction worked well for some arguments, such as title, vulnerability, and vendor, but did not perform well for other attributes, such as the product names. The low score for the product name extraction is caused by a high false positive rate, as many product names were not recognized. Besides the extraction quality, the evaluation also shows that not all advisories contained all attributes to begin with. Especially, the attributes of vulnerability, severity, and score were not provided for several advisory documents. Some of the missing attributes can be ascribed to the advisories being focused on the release of a patch, rather than the publication of vulnerabilities. While the recognition worked well in general due to the prevalence of the CVE numbers, some vulnerabilities without CVE numbers were not extracted.

7 Discussion and Conclusion

Based on requirements from literature and interviews with security professionals, we proposed a system that automatically retrieves new security advisory documents and transforms them into a machine-readable format. In addition to standardizing advisories by transforming them into the CSAF file format, the system offers improvements regarding the different types of advisory sources that can be polled for new documents. These improvements result in an overall retrieval rate of over 90%. Furthermore, we found that the retriever classes have different characteristics regarding their configuration, the amount of advisory metadata they extract from the overview page, such as the publishing date, and their stability against changes in the structure of the retrieved overview page.

Table 5. Characteristics of the Retriever Classes

Retriever	Config	Metadata	Flexibility	Robustness
Table	Easy	Good	Average	Average
XPATH	Hard	Average	High	Low
URL	Easy	Low	High	Good
Ajax	Hard	Good	High	Low
RSS	Easy	Good	Low	Good
CSAF	Easy	Good	Low	Good

These insights are summarized in Table 5. Compared to the work of [9] and [5], vulnerabilities are processed from a wider variety of sources, including websites. This makes the system more robust compared to other systems that depend on individual aggregated sources, such as the NVD. The system offers an improvement compared to the work of [5], as the addition of new data sources was simplified. The extraction of metadata from overview pages, in addition to the extraction of data from the detail page content, is an improvement over Taranis NG, which only extracts data from monitored detail pages.

7.1 Implications for Design

By reflecting on the findings and identified user requirements of our study, we want to discuss the following five implications for designing systems for the retrieval and extraction of security advisory documents (RQ1).

Requirement Satisfaction for the Selection, Retrieval, and Extraction of Security Advisories (D1). For the processes of source selection, document retrieval, and data extraction, the non-functional requirements of data quality, data origin, automation, resilience, timeliness, timestampedness, standardization, configurability, and scalability were identified. The requirements of configurability and scalability were satisfied by the ability of the system to monitor hundreds of sites, that can either be added manually or imported via a CSV file. The system is more scalable than comparable systems since it avoids using headless browsers but instead directly retrieves data from the back-end APIs of the providers, thus saving both bandwidth and processing time. The requirement for automation is only partially met by the source selection process since the recommendation of new advisory sources is only semi-automatic and still needs manual work. The user needs to manually evaluate whether the recommended site is suitable for ingestion, find the corresponding overview site, and configure the correct retriever for the recommended provider. The overview and detail page monitoring fulfilled the requirements of configurability, automation, and timeliness since the user can configure arbitrary sites to be monitored, and new advisories are automatically detected shortly after they have been published. Resilience against changes and data quality was challenging for the data extraction step since the extractor did not perform well for some attributes, such as product names. While the data origin principle was maintained on a document level, since each created CSAF contains a reference to the URL from which it was retrieved, maintaining data origin on an attribute level was not possible. The CSAF format, which was used as a central data structure for security advisory documents, does not allow storing metadata on how exactly attributes were extracted from a document.

Lack of Support for Automatic Retrieval (D2). While the advisory providers did not take any measures to hinder non-human access to their published documents, they also did not take any significant measures to support automatic retrieval. Many of the overview pages were JavaScript-dependent, effectively locking out non-human parties such as search machine indexing agents. The effect of that was somewhat mitigated by the providers offering RSS feeds, but the majority (70%) of sites still did not offer such a service. Since the system of [9] solely depends on RSS for retrieval of security advisory documents, the developed system can retrieve a significantly larger share of advisory documents. To compensate for the missing advisory documents, [9] also ingested data from the NIST National Vulnerability Database (NVD). However, this approach is considered risky due to the NVD being a single point of failure, compared to the decentralized nature of different individual vendors. That the risk associated with centralization is not unfounded is demonstrated by the fact that there is a significant backlog of unenriched vulnerabilities in the NVD as of April 2024 [18]. While 30% of the surveyed sites offered security advisory information as an RSS feed, only 8% of the sites offered a feed in the CSAF format, once again showing the difficulties of the standard being adopted and thus justifying the importance of this research.

Issues with Data Origin and Uncertainty (D3). The CSAF format cannot preserve data origins on an attribute level, since no information can be stored on how a particular value was extracted from a document. For instance, in the implementation, multiple approaches were taken for the extraction of CVSS scores: They were either taken from the metadata displayed on the overview page, extracted directly from the text content of the detail page, or calculated from extracted CVSS vectors. In the interviews, it was mentioned that the data can only be trusted if the system is transparent about how the presented information was obtained. This is especially true once conventional extraction methods are mixed with machine learning-based extraction methods. Similarly, the CSAF format was designed to represent absolute and not probabilistic information. For example, one may want to use multiple extraction techniques and then store their results together with a probability that the extracted information is correct. The current architecture only supports this up to the point where a CSAF file is created, and the system needs to decide on which results to accept or discard. However, if the probabilities and alternative values were kept, they could later be used during the correlation step to cross-check them with data from other providers and resolve conflicting information better. The concept of probability extraction and attribute-level data origin could be implemented by changing the data structure so that instead of atomic values, a 3-tuple is inserted into the CSAF scaffold. This 3-tuple would contain the attribute's value, the probability of it being correct, and the information on how it was derived from the source document. Alternatively, the whole CSAF could be wrapped in a dictionary and metadata could be stored alongside the CSAF document in a subdictionary.

Advisory Documents as CSA Objects (D4). While existing approaches have been applying the CSA concept to cyber defence [1], not much work has been done on how security advisory documents fit into the picture. This work presents a model where security advisory documents and the referenced vulnerabilities correspond to the elements of [3]. While the notion of "time" can be interpreted as the timestamps at which an advisory document was published or changed, the notion of "space" can be interpreted as the site on which the information was published, as well the location of the document within a network of documents interlinked by references. By combining the timeline of the published documents and the extracted references to other documents, one can model the flow of information between the retrieved sources. While this has already been done in academic settings for a limited number of sites, as in the work of [16], there exist no systems that present users with this information, allowing them to make projections. For example, if multiple independent advisory providers reference the same advisory in quick succession, this may be interpreted as a signal for a high-severity vulnerability. Thus, to make temporal and spatial aspects of security advisories understandable to humans, vulnerability-centered CSA systems should make use of timeline graphs and network graphs. For example, the visualization of timelines could help the user to make projections on how the risk of exploitation will evolve over time [8].

Vulnerability Front-Running and Ethical Concerns (D5). According to our interviewees, quicker access to information is the main reason why they invest a significant amount of time in directly retrieving advisory documents from vendors instead of solely relying on the feed of the upstream CERT. This raises the question of whether adversaries could use the developed system too, as a form of *vulnerability front-running*, to gain an advantage over CERTs not using such a system but instead relying on an aggregated feed. The concept of front-running is taken from the field of finance, where dishonest participants use non-public information on an impending market-changing transaction to their own advantage. The problem of vulnerability front-running is not only exacerbated by the delay between initial publication and publication in aggregated feeds, which can be observed in the case of the NVD but also in the decreasing time between the publication of vulnerabilities and their exploitation [18]. The work of [4], which uses Large Language Models (LLMs) to generate exploits for new vulnerabilities automatically, shows that we can soon expect publication-to-exploitation timeframes of a few seconds. This development shows that a CSA system should help humans make better and quicker decisions. These developments may lead to a technological arms race between CERTs and adversaries to retrieve and interpret advisory documents as quickly as possible, ultimately questioning the role of the human *in* the advisory processing loop, or in advisory providers taking steps against the automatic retrieval of advisory documents, such as CAPTCHAs.

7.2 Limitations and Future Work

Overall, since the interviews touched upon multiple topics, it cannot be guaranteed that the process model covers all vulnerability-handling activities and more empirical research, such as cognitive walkthroughs of the vulnerability management processes, should be conducted on how CERTs gain CSA from security advisories. Furthermore, limited resources were spent on the development of the preliminary system, especially the extraction component. Due to the high variability of content on detail pages, more effort is needed to develop a robust extraction solution that can gracefully handle the many potential edge cases. In particular, the extraction and standardization of vendor and product names should receive more attention, since having a high level of accuracy in extracting these arguments is essential for later processing steps such as correlation and filtering. Lastly, only the first three steps of the processing pipeline were implemented. Thus, cognitive aspects, which play an essential role in CSA, were not yet examined. This work focused on the base processes required for establishing vulnerability-centred CSA but did not answer the question of how the retrieved and extracted data should be best presented to enhance CSA. For this reason, the subsequent steps of the processing pipeline should also be implemented, ideally informed by a user-centred design process which iteratively incorporates user evaluations to assess the acceptance, functionality, and usability of the system.

Acknowledgments. This research work has been co-funded by the German Federal Ministry of Education and Research (BMBF) and the Hessian Ministry of Higher Education, Research, Science and the Arts (HMWK) within their joint support of the National Research Center for Applied Cybersecurity ATHENE, as well as by the BMBF in the project CYLENCE (13N16636). A preliminary version of this study was presented at a workshop [11], which did not comprise the (I) results of our empirical pre-study, (II) discussion of non-functional requirements, (III) extraction and standardization evaluation, nor a (IV) thorough discussion of design implications.

References

1. Albanese, M., et al.: Computer-aided human centric cyber situation awareness. In: Theory and Models for Cyber Situation Awareness, pp. 3–25 (2017). https://doi.org/10.1007/978-3-319-61152-5_1
2. Braun, V., Clarke, V.: Using thematic analysis in psychology. Qual. Res. Psychol. **3**(2), 77–101 (2006). https://doi.org/10.1191/1478088706qp063oa
3. Endsley, M.R.: Situation awareness global assessment technique (SAGAT). In: Proceedings of the IEEE 1988 National Aerospace and Electronics Conference, pp. 789–795. IEEE (1988). https://ieeexplore.ieee.org/abstract/document/195097/
4. Fang, R., Bindu, R., Gupta, A., Kang, D.: LLM Agents can Autonomously Exploit One-day Vulnerabilities (2024). http://arxiv.org/abs/2404.08144. arXiv:2404.08144
5. Fenz, S., Ekelhart, A., Weippl, E.: Fortification of IT security by automatic security advisory processing. In: 22nd International Conference on Advanced Information Networking and Applications (AINA 2008), pp. 575–582. IEEE (2008). https://ieeexplore.ieee.org/abstract/document/4482758/
6. Franke, U., Brynielsson, J.: Cyber situational awareness - a systematic review of the literature. Comput. Secur. **46**, 18–31 (2014). https://doi.org/10.1016/j.cose.2014.06.008
7. Husák, M., Jirsík, T., Yang, S.J.: SoK: contemporary issues and challenges to enable cyber situational awareness for network security. In: Proceedings of the 15th International Conference on Availability, Reliability and Security, pp. 1–10 (2020). https://doi.org/10.1145/3407023.3407062
8. Jiang, L., Jayatilaka, A., Nasim, M., Grobler, M., Zahedi, M., Babar, M.A.: Systematic literature review on cyber situational awareness visualizations. IEEE Access **10**, 57525–57554 (2022). https://doi.org/10.1109/ACCESS.2022.3178195
9. Kaufhold, M.A., Riebe, T., Bayer, M., Reuter, C.: 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 Conference on Human Factors in Computing Systems (CHI), Honolulu, HI, USA (2024). https://doi.org/10.1145/3613904.3642368
10. Kaufhold, M.A., Bäumler, J., Bajorski, M., Reuter, C.: Cyber threat awareness, protective measures and communication preferences in Germany: implications from three representative surveys (2021–2024). In: Proceedings of the Conference on Human Factors in Computing Systems (CHI), CHI 2025, Yokohama, Japan. Association for Computing Machinery (2025). https://doi.org/10.1145/3706598.3713795
11. Kaufhold, M.A., Bäumler, J., Koukal, N., Reuter, C.: Towards a security advisory content retrieval and extraction system for computer emergency response teams. In: Mensch und Computer 2024 - Workshopband. Gesellschaft für Informatik e.V., Karlsruhe, Germany (2024). https://doi.org/10.18420/muc2024-mci-ws13-133
12. Kaufhold, M.A., Bäumler, J., Koukal, N., Reuter, C.: GitHub: Appendix and Source Code of the AdvisoryHub Application (2025). https://github.com/PEASEC/advisory-hub
13. Lekkas, D., Spinellis, D.: Handling and reporting security advisories: a scorecard approach. IEEE Secur. Priv. **3**(4), 32–41 (2005). https://doi.org/10.1109/MSP.2005.98
14. Mancuso, V., McGuire, S., Staheli, D.: Human centered cyber situation awareness. In: Ahram, T., Karwowski, W. (eds.) AHFE 2019. AISC, vol. 960, pp. 69–78. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-20488-4_7

15. Mayring, P.: Qualitative content analysis. Forum Qual. Soc. Res. **1**(2) (2000). https://doi.org/10.17169/fqs-1.2.1089
16. Miranda, L., et al.: On the flow of software security advisories. IEEE Trans. Netw. Serv. Manag. **18**(2), 1305–1320 (2021). https://ieeexplore.ieee.org/abstract/document/9427134/
17. Morrell, N.E.: Quality function deployment. SAE Trans. 1090–1097 (1987). https://www.jstor.org/stable/44470091
18. Munshaw, J.: What's the deal with the massive backlog of vulnerabilities at the NVD? (2024). https://blog.talosintelligence.com/nvd-vulnerability-backlog-the-need-to-know/
19. Overmeer, M.: Taranis3 Documentation (2018). https://github.com/markov2/taranis3
20. 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. Proc. ACM Hum. Comput. Interact. (PACM) Comput.-Supported Coop. Work Soc. Comput. **5** (2021). https://doi.org/10.1145/3479865
21. Skopik, F., Akhras, B.: Taranis AI: applying natural language processing for advanced open-source intelligence analysis. ERCIM News **2024**(136), 50–51 (2024). https://ercim-news.ercim.eu/en136/r-i/taranis-ai-applying-natural-language-processing-for-advanced-open-source-intelligence-analysis
22. Venable, J., Pries-Heje, J., Baskerville, R.: A comprehensive framework for evaluation in design science research. In: Peffers, K., Rothenberger, M., Kuechler, B. (eds.) DESRIST 2012. LNCS, vol. 7286, pp. 423–438. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29863-9_31
23. 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, pp. 187–199 (2024). https://doi.org/10.1145/3688459.3688463