On Collaboration and Automation in the Context of Threat Detection and Response with Privacy-Preserving Features

LASSE NITZ and MEHDI AKBARI GURABI, Fraunhofer FIT, Sankt Augustin, Germany and RWTH Aachen University, Aachen, Germany
MILAN CERMAK, Masaryk University, Brno, Czech Republic
MARTIN ZADNIK, CESNET, Prague, Czech Republic
DAVID KARPUK, W/Intelligence, WithSecure Corporation, Helsinki, Finland
ARTHUR DRICHEL, SEBASTIAN SCHÄFER, and BENEDIKT HOLMES, RWTH Aachen University, Aachen, Germany
AVIKARSHA MANDAL, Fraunhofer FIT, Sankt Augustin, Germany

Organizations and their security operation centers often struggle to detect and respond effectively to an extensive quantity of ever-evolving cyberattacks. While collaboration, such as threat intelligence sharing between security teams, and response automation are often discussed in the cybersecurity community, issues like data sensitivity and confidence in detection may hinder their adoption. This work investigates the potentials and challenges of collaboration and automation to enhance incident response processes. We propose a reference architecture for data sharing in threat detection and response, aiming to boost collaborative and automated efforts across organizations while also considering privacy-preserving features. To address these challenges and potentials, we discuss how such a framework could enhance current response processes within and between organizations, validated with results in local attack detection, incident response, and data sharing.

CCS Concepts: • **Security and privacy → Malware and its mitigation**; **Intrusion detection systems**; *Data anonymization and sanitization*; *Privacy-preserving protocols*; • **Information systems → Collaborative and social computing systems and tools**; • **Computing methodologies →** *Machine learning*; • **Human-centered computing →** *Visualization*;

Additional Key Words and Phrases: Cybersecurity, collaborative detection and response, incident response automation, information sharing, privacy

This work has received funding from the EU H2020 LEIT Information and Communication Technologies programme for the project SAPPAN (Sharing and Automation for Privacy Preserving Attack Neutralization) under grant agreement No. 833418.

Authors' Contact Information: Lasse Nitz (corresponding author), Fraunhofer FIT, Sankt Augustin, Germany and RWTH Aachen University, Aachen, Germany; e-mail: lasse.nitz@fit.fraunhofer.de; Mehdi Akbari Gurabi, Fraunhofer FIT, Sankt Augustin, Germany and RWTH Aachen University, Aachen, Germany; e-mail: mehdi.akbari.gurabi@fit.fraunhofer.de; Milan Cermak, Masaryk University, Brno, Czech Republic; e-mail: cermak@ics.muni.cz; Martin Zadnik, CESNET, Prague, Czech Republic; e-mail: zadnik@cesnet.cz; David Karpuk, W/Intelligence, WithSecure Corporation, Helsinki, Finland; e-mail: davekarpuk@gmail.com; Arthur Drichel, RWTH Aachen University, Aachen, Germany; e-mail: drichel@itsec.rwth-aachen.de; Sebastian Schäfer, RWTH Aachen University, Aachen, Germany; e-mail: schaefer@itsec.rwth-aachen.de; Benedikt Holmes, RWTH Aachen University, Aachen, Germany; e-mail: holmes@itsec.rwth-aachen.de; Avikarsha Mandal, Fraunhofer FIT, Sankt Augustin, Germany; e-mail: avikarsha.mandal@fit.fraunhofer.de.

ACM 2576-5337/2025/2-ART5
https://doi.org/10.1145/3707651

ACM Reference format:
Lasse Nitz, Mehdi Akbari Gurabi, Milan Cermak, Martin Zadnik, David Karpuk, Arthur Drichel, Sebastian Schäfer, Benedikt Holmes, and Avikarsha Mandal. 2025. On Collaboration and Automation in the Context of Threat Detection and Response with Privacy-Preserving Features. *Digit. Threat. Res. Pract.* 6, 1, Article 5 (February 2025), 36 pages.
https://doi.org/10.1145/3707651

1 Introduction

In the past few years, there has been a significant rise in cyberattacks across various sectors impacting our society and economy. According to the **European Union (EU)** Agency for Network and Information Security Threat Landscape Report 2023 [38], there has been a more than 150% increase in globally reported incidents from the last half of 2022 to the first half of 2023. Ransomware and malware threats are on the rise, where cybercriminals are effectively utilizing ransomware/malware-as-a-service and purchasing sophisticated malware from underground sources to launch targeted attacks [38]. Among the ongoing political tensions in regions like Russia-Ukraine and Israel-Palestine, threat actors are targeting critical infrastructures such as the energy and defense sector with malware and social engineering-based attacks [65].

This rapidly evolving threat landscape presents a substantial challenge for many organizations. Cybersecurity incidents, also caused by advanced persistent threats, are omnipresent and call for rapid and effective detection and response [9, 26]. Hence, organizations must employ effective threat detection and response processes starting from understanding their technical and organizational environment to appropriately responding to cyberattacks and recovering from their consequences [87]. Moreover, small- or mid-sized enterprises do not have sufficient resources or expertise to face advanced threats [20]. Thus, they need to rely on outsourcing threat detection and response to external cybersecurity service providers. In such cases, the privacy of user data and the confidentiality of an organization's business information become major concerns while outsourcing.

Additionally, **Security Operations Centers (SOCs)**, Computer Emergency Response Teams, and **Computer Security Incident Response Teams (CSIRTs)** are grappling with a significant shortage of skilled personnel [96]. SOC analysts often face burnout while manually filtering out a large number of false-positive alerts. Given the sheer volume of false-positive alerts, alerts caused by actually malicious activities may go unnoticed and undetected amid background noise, thereby increasing the risk of damage to organizational assets [7]. In intrusion detection, rule-based and signature-based approaches suffer from identifying new types of attacks with unseen patterns, whereas, **Machine Learning (ML)**-based approaches tend to excel at identifying new types of threats, and detecting anomalies. However, the lack of interpretability in ML models may hinder their usage, as analysts may find decisions from rule-based approaches more understandable while investigating false alarms [46]. In the case of correctly triggered incidents, SOC operators manually initiate specific response actions based on factors such as the type of detected attack, the confidence level linked to the detection, the significance of the asset, and the evaluation of the incident's severity. The response actions can be quite repetitive and performing them manually may lengthen the response time. Therefore, recommending suitable response actions or even automatically executing common response actions for certain types of attacks based on their severity could significantly enhance the response workflow in SOCs.

While addressing the aforementioned challenges, SOC teams may immensely benefit from collectively and collaboratively sharing **Cyber Threat Intelligence (CTI)** [18, 19]. CTI encompasses a wide range of information, whether tactical, strategic, or operational, designed to detect and handle incidents [60]. Recent work from Preuveneers and Joosen [82] pointed out that CTI such as traditional **Indicators of Compromise (IoCs)** could lose its impact over time due to its volatile nature (e.g., the IP address of a compromised machine may change) and proposed an access control-based sharing scheme of ML models for threat detection. Sharing ML-based detection models as the new-generation IoCs has the potential to help organizations detect and respond to new types of attacks on time.

Subsequently, privacy issues need to be handled carefully before or during the sharing of CTI between organizations, especially since CTI may include personal information about end users, confidential company information, and sensitive infrastructure details. Organizations could collaboratively train threat detection models on incident data by utilizing **Privacy-Enhancing Technologies (PETs)** such as federated learning, data sanitization for pre-processing, or cryptographic techniques. Especially federated ML techniques can facilitate collaborative attack detection between organizations and could improve the incident detection accuracy by using shared global ML models, e.g., for the purpose of anomaly or malware detection. By collaboration, one could not only reduce biases in the training set but may also achieve higher accuracy and reduced false positives. Organizations could also benefit from sharing cybersecurity playbooks [84], which provide step-by-step guidelines for response to and recovery from different types of incidents. However, cybersecurity playbooks are often organization-specific and are documented in unstructured or semi-structured formats [3]. Thus, there is a demand for standardized, machine-readable, interoperable playbook formats to facilitate sharing and seamless integration into the local system, which could enable further automation.

Improving an organization's incident response capabilities and collaborations are also important from a legal perspective, not only due to privacy laws [8] but also with emerging cybersecurity regulations. For example, in early 2023, the EU approved a new version of the **Network and Information Systems Directive 2 (NIS2)** [86]. NIS2 defines the EU's cybersecurity priorities for enhanced security and resilience in organizations against evolving threats. To comply with such legislation, companies must prepare themselves to meet several cybersecurity requirements such as following certain incident response procedures (e.g., prompt incident reporting to national authorities). In case of non-compliance, companies may have to pay hefty fines.

Our Contributions. In this article, we contribute to various aspects of threat detection and incident response, which can be presented fourfold. First, we identify key potentials and challenges of privacy-preserving collaboration and automation in threat detection and incident response. Our insights are derived from a practical perspective, based on our experiences gained in the EU H2020 cybersecurity project named **Sharing and Automation for Privacy-Preserving Attack Neutralization (SAPPAN)**.[1] The project comprised of a consortium with 77 individuals from 8 organizations, involving security providers, CSIRTs, corporate entities, and academic partners. Second, we propose a reference architecture to enable CTI sharing between organizations. The architecture includes different technical components and their interactions to facilitate privacy-preserving sharing of CTI and response automation in a generic organization that wants to participate in collaboration. Third, we present an extensive array of results in threat detection and response, clustered into three tasks: local attack detection, incident response, and sharing. The local attack detection task surveys our ML-based approaches to improve malware detection and anomaly detection. The incident response task contributes to three topics: cybersecurity playbook management, incident similarity models for recommendation and support of human operators, and incident response automation by demonstrating an automated malware analysis workflow. In the sharing task, we contribute to privacy-preserving collaboration such as federated approaches for threat detection, data sanitization approaches for sharing, and cybersecurity playbook sharing. Fourth, drawing from our validation results, we retrospectively discuss the efficacy and limitations of different results by mapping them to relevant challenges and potentials, and aligning them with the proposed reference architecture. We further outline directions for future research.

Outline. The remainder of this work is structured as follows: Section 2 provides essential background on the incident response lifecycle. Section 3 presents our identified list of key potentials and challenges of collaboration and automation when adapted to the incident response lifecycle. Section 4 proposes our framework and presents a reference architecture for a general incident detection and response system, covering a description of individual technical components and their interactions. Sections 5–7 cover our approaches and developed components for the tasks of local attack detection, incident response, and sharing, respectively. Section 8 discusses our evaluation

[1] SAPPAN on CORDIS: https://doi.org/10.3030/833418; SAPPAN project Web site: https://sappan-project.eu/

results for each task, validated with technically quantifiable metrics as well as opinion-based perspectives. Section 9 retroactively discusses our results from the perspectives of challenges and potentials, as well as the proposed reference architecture. Lastly, Section 10 concludes this work and provides directions for future research.

2 Background

Organizations vary in the maturity of their detection and response capabilities, and they also interpret incidents differently, thus making it difficult to establish unified response strategies [26]. An incident response process model can help organizations to plan their response strategies in a standardized way. There exist several process models for incident responses. Two of the prominent ones are the **National Institute of Standards and Technology (NIST)** incident response lifecycle [25], and the SANS Institute's PICERL model [74]. Most of the frameworks present similar stages of response with slight variations in wording and details. To understand the common response stages, we briefly discuss the NIST incident response lifecycle, which consists of four main phases [25].

The *Preparation* phase includes building up the IT infrastructure, software, and hardware for the response team, as well as ensuring the functionality of monitoring tools and reporting mechanisms. During this phase, organizations aim to minimize incidents by implementing controls such as risk assessment and malware protection. Additionally, organizations increase awareness and provide training to their analysts to improve incident handling.

In the *Detection and Analysis* phase, incident data is collected to detect past/ongoing incidents (indicators) or potential future incidents (precursors). Sources for detecting indicators or precursors include **Intrusion Detection/Prevention Systems (IDS/IPS)**, antivirus and antispam software, application and network logs, and publicly available information such as the National Vulnerability Database. While most IDS/IPS products, commonly utilizing attack signatures or statistical detection methods to identify unusual activities, produce alerts with relevant data, they often generate false positives, requiring manual analysis and validation of alerts. Moreover, incidents are assessed and prioritized based on their impact. There are also obligations to notify affected entities according to policies and legal frameworks (e.g., NIS2 mandates an initial report to be submitted within 24 hours [86]).

During the *Containment, Eradication, and Recovery* phase, the core incident response activities take place. The containment part includes essential decision-making (e.g., whether to initiate a system shutdown or to isolate parts of a network) based on pre-defined strategies or playbooks, depending on acceptable risks to avoid further damage. Forensic techniques are also employed to gather evidence, simultaneously documenting logs for resolving the issue and meeting legal requirements. When the incident is contained, analysts focus on eradication and recovery, aiming to restore system operations to a normal state. This may involve tasks such as restoring backups, rebuilding systems, applying patches, and increasing security measures to prevent similar incidents.

The *Post-Incident Activities* phase includes lessons learned, and reviews with all involved parties after major incidents to improve future incident response. This retrospective phase initiates technical measures, e.g., by evaluating whether collected incident data can be utilized in the preparation phase for purposes such as risk assessment or implementation of additional controls.

Our Approach. Figure 1 presents our collaboration approach for threat detection and response aligned with the NIST incident response framework. In the local view, different phases of the incident response lifecycle take place within an organization. In the collaborative view, CTI is shared from the local view in different phases of the lifecycle to establish a common understanding of attack detection, incident assessment, and handling. The privacy issues are handled before or during sharing of CTI. Moreover, the processes can be automated at any phase of the response lifecycle. However, in this work, we only investigate automating processes within response activities, specifically focusing on the *Containment, Eradication, and Recovery* phase, as illustrated in the local view in Figure 1.

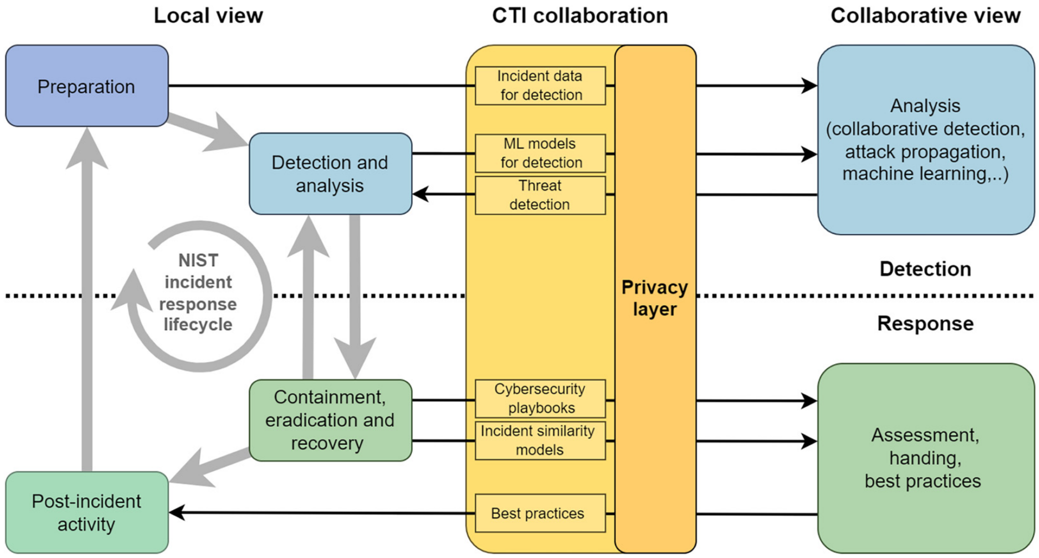

Fig. 1. SAPPAN collaboration concept aligned with the National Institute of Standards and Technology (NIST) incident response lifecycle.

3 Key Potentials and Challenges

To gain a better understanding of how collaboration and automation can affect work in the cybersecurity domain, we have identified key potentials that they may bring as well as key challenges that need to be overcome by respective solutions. For this, we decided to take a 2-step divergence-convergence approach, which first creatively explores different options and then analytically condenses the initial findings down to a refined result. This reflects the problem identification phase (consisting of a discovery and a definition step) of the double diamond model, a popular process model in design thinking (see, e.g., [73]). A total of 17 experts of the SAPPAN consortium participated directly in the identification of key potentials and challenges, which allowed to capture a great variety of different perspectives from both industry and academia. The experts' backgrounds cover the areas of threat detection, incident handling, security services and consultation, ML and data science, PETs, and data visualization in the cybersecurity domain.

In the first step, each participating expert has been asked to think of and document possible potentials and challenges of integrating collaboration and automation into cybersecurity use cases. This part of the process did not include active exchange between the different participants. For the second step, we have organized an interactive workshop with the goal of condensing the collected potentials and challenges down to the key ones. To achieve this, the participants have first been tasked with identifying clusters in the collected potentials and challenges. Subsequently, the clusters have been discussed. For this part of the process, the goal was to summarize the points within a cluster, and to define a generalized description for each cluster. Afterwards, the cluster descriptions were checked against each other to identify possible overlaps or conflicts.

This process has been carried out separately for collaboration and automation. The resulting key potentials and challenges are listed in Table 1 for collaboration and Table 2 for automation. These key potentials and challenges are further used as guidance for the discussion of the validation results as given in Section 9.

While our identification of challenges and potentials primarily focuses on technical aspects, recent work has also looked at challenges and potentials from the perspective of interrelationships of technical and social aspects in CTI sharing communities [45]. As such, our work complements these findings from a technical point of view.

Table 1. Challenges and Potentials in Collaboration

ID	Name	Description
Collaboration challenges		
C.C1	Sensitive information	Concerns about leaking sensitive information include customer privacy violations resulting in reputation damage, organizational vulnerabilities exposed to adversaries, and regulatory non-compliance.
C.C2	Machine-interpretable data	Machine-interpretable data is essential for enabling effective collaboration on a larger scale. It is, for example, important for finding relevant data, assessing the quality of shared data, and using shared data in a plug-and-play manner.
C.C3	Data model universality	The data model and its format must be widely accepted to enable the exchange of data between a large variety of actors.
C.C4	Data model adaptability	The data model must be flexible enough to accommodate future extensions such as different types of IoCs, attack types, or response information.
C.C5	Quality assurance	The quality of exchanged information determines if a collaboration is beneficial or not. It includes multiple aspects such as data relevance, timeliness, completeness, noisy and false information.
Collaboration potentials		
C.P1	Preemptive defense	Preparation against novel kinds of attacks, even before they were observed in the own organization by sharing data identifying the attack or the attack source (learning from others' experiences).
C.P2	Collaborative data generation and learning	Collaborative creation of balanced datasets and collaborative learning. The supervised ML-based detection of malicious behavior requires balanced and diversified datasets to reduce false positives, which may benefit from sharing information about the availability of datasets and their rating, or collaborative/federated ML operating over the sharing platform.
C.P3	Standardization	Standardized description to enable machine understanding as well as wide deployment.
C.P4	Validation via visualization	Increased trust in data/models through interpretability, even when considering large amounts of data, complex problem domains, or privacy concerns.

4 The SAPPAN Reference Architecture

As the key potentials and challenges identified in the previous section partly result from interactions of different technical components of a sharing-enabled incident response system, it is important to understand the context in which these components interact. For this purpose, we propose a general architecture, which we describe further below and refer to as the SAPPAN reference architecture. A visual representation of this architecture is given in Figure 2.

To align the different components of the SAPPAN reference architecture with different purposes as well as different stages in the incident response life cycle, we cluster these components into three tasks: local attack detection, incident response, and sharing. In the following, we briefly introduce the purpose of each component of the architecture in accordance with this clustering. We further utilize the clustering into these three tasks in subsequent sections to discuss the background and our contributions to each of them. The SAPPAN reference architecture is also revisited in Section 9 to discuss the implications of our validation results concerning the architecture and its components.

Table 2. Challenges and Potentials in Automation

ID	Name	Description
Automation challenges		
A.C1	Risk of disruption	The risk of disruptions, where ill-defined procedures might cause damages.
A.C2	Implementation effort	Implementing automation demands weighing costs against benefits. Identifying appropriate scenarios and integrating them with existing systems is essential.
A.C3	Nuances of incidents	Incidents' nuances and their complex nature pose challenges, particularly in decision-making for branching within automated workflows as we transition from general to automated playbooks.
A.C4	Confidence in automated decisions	Confidence in automation decisions and ensuring good data quality is foundational for the success of the automated response.
Automation potentials		
A.P1	Workload reduction	Automation has the potential to substantially reduce the workload of human operators. Handling routine incidents might allow human operators to focus on more complex/unusual incidents and mitigate alert fatigue.
A.P2	Faster processing and resolution of incidents	Timely incident processing and resolution are potential benefits of automation. Associated potential benefits are its scalability, non-reliance on working hours, and trimmed operational costs.
A.P3	Integrated logging	Integrated logging in automation ensures every action is tracked, simplifying statistics generation and negating potential employee surveillance concerns. If no humans are involved, employee rights do not have to be considered, which might otherwise be a concern.
A.P4	Increased quality via human–machine teaming	Human–machine teaming carries the potential to augment the quality of ML models. Visual analytics can further work as a gap-filler for non-automatable parts in response and recovery and can be used for monitoring the system as well as recommendation and validation purposes.

4.1 Local Attack Detection

The *system and network monitoring* component takes care of all activities related to observing ongoing activities. Its primary purpose within the SAPPAN reference architecture is to collect data, which is generated as part of system monitoring. This data is then shared with components that require such data as input, e.g., to create detection system components or to detect malicious activities within live traffic.

The *ML* component is intended to train models and coordinate activities related to collaborative efforts in ML, such as federated learning. It provides trained models for the detection system and the ML visualization component.

The component for *ML visualization* aims to support human users to better understand the trained models by providing suitable visualizations.

For the purpose of detecting incidents in the monitored system, respective monitoring data is provided to the *detection and analysis system*. In this context, trained classifiers can be utilized to detect malicious samples or patterns in the data provided to the detection and analysis system.

Monitoring data is further provided to a *profiling* component, which creates profiles based on traffic observed in the past. The purpose of these profiles is to capture the usual behavior of monitored system actors (such as hosts or applications), which can further be utilized for detecting anomalous behavior on new monitoring data.

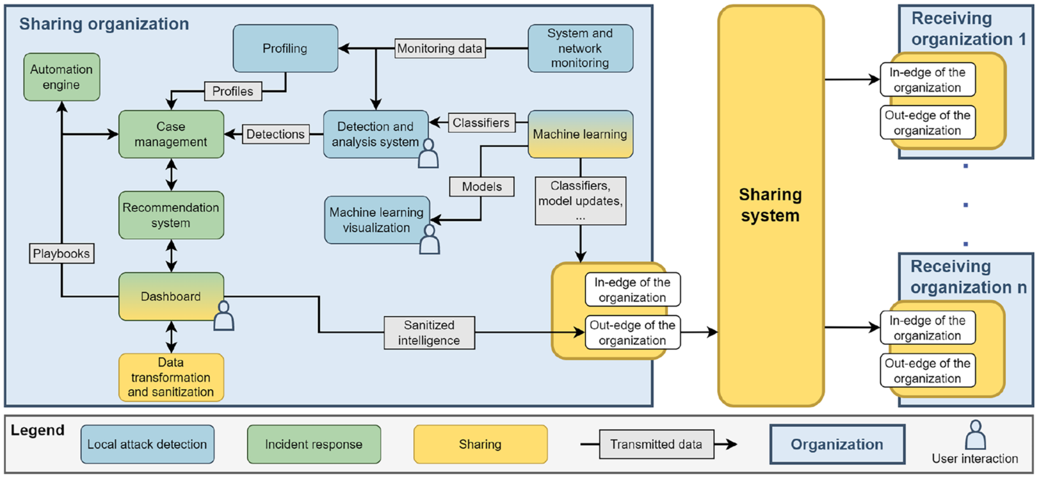

Fig. 2. Graphical representation of the SAPPAN reference architecture. Different technical components are represented as nodes and interactions between them as edges. Components with user interactions are marked respectively. Note that this figure depicts the sharing flow of an organization. When receiving data, some of the edges change direction.

4.2 Incident Response

The *case management* component uses profiles and detections by the respective components to organize activities related to new incidents, actions taken, and the archiving of old incidents.

The *recommendation system* analyzes either how specific cases have been handled in the past to recommend actions for similar cases to a human operator or implements its own heuristics to infer the next actions.

The *dashboard* aims to guide a human user through the incident handling process and further provides the interface to communicate with the sharing system. The dashboard is a user interface to access and interact with the systems to benefit from past cases through recommendations. Actions taken by the operator can further be utilized for future recommendations.

In order to reduce the manual work for human operators, the *automation engine* is intended to automatically resolve frequent low-impact incidents. This requires initial implementation efforts, in which cybersecurity playbooks are specified into executable source code, which interacts with existing systems that are deployed within the organization.

4.3 Sharing

Since data sharing introduces requirements surrounding normalization and data protection, a *data transformation and sanitization* component is connected to the dashboard. A human operator can apply sanitization and normalization operations there to comply with respective expectations and obligations before making data accessible to external parties.

Because the human operator interacts with the data transformation and sanitization component via the *dashboard*, the option to initialize data sharing after the data transformation and sanitization process is also included in the dashboard's functionality. To provide the human operator with a user interface for coordinating the interaction with the sharing system, the dashboard is connected to the in- and out-edge of the organization.

Also the *ML* component has to deal with aspects of sharing, for example, to exchange model updates with collaborators in federated learning tasks. As such processes do not require that model updates be exchanged manually by a human operator, the ML component is directly connected to the sharing system via the organization's in- and out-edges.

While not necessarily part of any of a sharing/receiving organization's infrastructure, the *sharing system* connects the different organizations and ensures that shared data is distributed to the intended recipients. As such, it is connected to the in- and out-edges of the participating organizations, which provide an interface between the sharing system and an organization's internal systems.

5 Task: Local Attack Detection

The first step in responding to an incident and ensuring the security of a network and its services is detecting and analyzing intrusions at the local network level [27]. This can be challenging for security specialists due to a large amount of diverse data and the need for the detection of various threats while reducing false positives. In the cybersecurity domain, the following three data groups based on their origin are typically used: network-based data (observed in a computer network), endpoint data (observed in computers and similar devices), and open source intelligence (cyberattack-related data that can be retrieved from other, usually publicly available, sources) [55]. Effective processing and analysis of such data then directly affect threat detection and timely mitigation capabilities. The anomaly and intrusion detection domain is a broad research area, and many different techniques and approaches emerged through the years [41]. In general, these techniques may be categorized into categories: statistical (time-series, univariable, etc.), ML, and others (signatures, threshold values, etc.) [99]. It is not possible to unambiguously determine which technique fits for what purpose, since factors such as the type of infrastructure, type of processed data, speed of data processing, the accuracy of the method (**False-Positive Rate (FPR)** and false-negative rate), or its universality, for example, need to be taken into account [21]. The key element is also an integration of these techniques into the incident response workflow and overall architecture, as described in Section 4.

Our research focused on two areas representing different analytical approaches facing challenges of local attack detection. The first is malicious domain detection, which is associated with malware and phishing attacks, representing today's main threats [57]. The second challenge we addressed is anomalous behavior detection both at the network and host levels. Our goal is to detect compromised devices even in the case of advanced, previously unknown attacks. These areas and their corresponding components are shown in Figure 2 in dark blue. The proposed methods can operate independently, but their main strength is gained when properly connected to data sharing and other incident response phases, as described in the following Task-related sections.

5.1 Malicious Domains Detection

Regarding phishing attacks, we focused on analyzing domains in **Certificate Transparency (CT)** logs, allowing us to obtain information about new domains quickly. In our research [31], we presented and evaluated a system designed to identify phishing certificates within CT logs in real time, as they are appended. In addition to real-time detection, this system supports retrospective analyses and addresses the issue of missing reference data by providing a validation mechanism. Our modular and scalable system provides a robust platform for developing new certificate detection algorithms and allows their benchmarking against current methods. Moreover, it aids in the creation of authentic datasets that are useful for training various ML classifiers.

Besides phishing attacks, we have also focused on malware-related domains, specifically domains generated by **Domain Generation Algorithms (DGAs)** used to contact **Command and Control (C2)** servers. In our related study [34], we introduced two innovative classifiers based on **Residual Neural Networks (ResNets)**. Through a comprehensive comparison with other leading classifiers, we demonstrated the effectiveness of our models. Specifically, in two binary classification scenarios (distinguishing **Algorithmically Generated Domains (AGDs)** from known DGAs and identifying AGDs from unknown DGAs), we showed that our binary classifier not only achieves top-tier classification accuracy but also has the lowest FPR. Additionally, we confirmed the real-time performance of our classifiers, proving that our ResNet-based binary classifier effectively identifies samples from previously unknown DGAs.

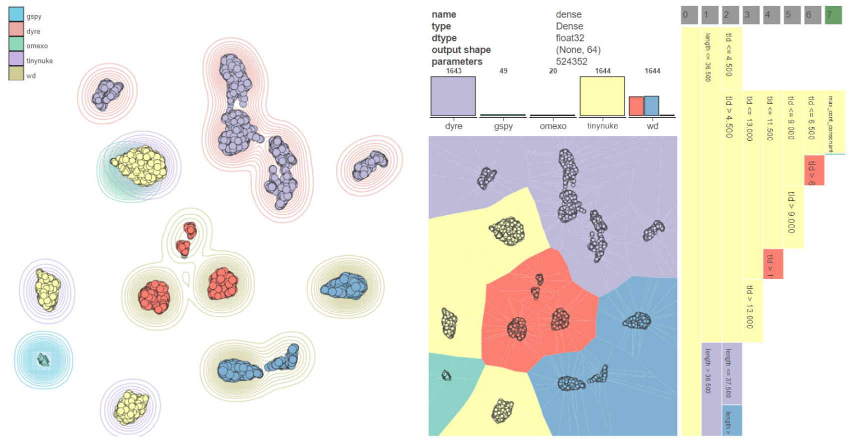

Fig. 3. The analysis view provides understandable interpretations of a DGA classification model [15].

To properly handle incident response, it is necessary to use existing detection methods and create new ones capable of responding to current threats. One relevant aspect in this context is the explainability of a newly developed model, which we have also focused on in our research [15]. We designed and tested a visual analytics system that enables designers of deep learning models for multi-class DGA classification to explore the patterns their models detect in the data. This allows users to analyze the progression of a customized subset of the dataset through the different layers of a deep learning model. We used clustering to identify patterns in the activations of the model's nodes and presented these patterns to the user with a decision tree for interpretation. An example of the analysis view is shown in Figure 3. Our evaluation of the system demonstrated how it can be used to better understand misclassifications, identify potential biases, and interpret the roles of different layers in a model.

5.2 Anomalous Behavior Detection

In general, intrusion detection methods can be divided into pattern-based and anomaly-based approaches. Pattern-based methods offer higher accuracy but may not be able to detect advanced or novel types of attacks. To address this issue, detection methods based on monitoring device behavior and identifying deviations from the normal state are used. These methods have a greater potential to identify even previously unknown types of attacks. Therefore, our research has focused on application and endpoint profiling serving as a basis for anomaly detection within the incident response context.

The goal of application profiling, or application fingerprinting, is to determine typical characteristics of an application, e.g., queried domains, as well as modeling the distinct behavior of an application in order to identify and classify the application. For application identification, we have proposed and evaluated rule-based process mining models that are able to identify whether certain applications are running on a device based on DNS traffic. This is, e.g., useful for network administrators or pentesters to get an overview of a network, but also for incident response handlers to get more context of the machine that produced the alert. For the application classification task, we have developed and successfully integrated into the SAPPAN architecture an approach to build process mining models based on system events. The idea is to build a model representing the typical behavior of an application in order to detect deviations, e.g., if the application is behaving maliciously.

Application behavior analysis also closely relates to analyzing processes and their relations aiming to detect anomalous events in endpoints. We have developed a model for identifying anomalous process launch events [56]. In order to increase the reliability of detections reported by the model and to support security analysts in handling those detections, we have experimented with combining detected anomalies in so-called provenance graphs. Further, we have developed a set of models designed to detect specific classes of anomalous endpoint behavior and a method for presenting connections among detected anomalies as a node-edge graph supporting decisions by incident investigators. We further supported their decision-making by creating an endpoint profiling tool, based on our 1-year study [53], that allows the analyst to classify hosts in the network and determine their expected behavior.

6 Task: Incident Response

Following the detection of an incident, an adequate response needs to be taken. However, the increasing volume of security data introduces challenges for organizations, as processing it requires more resources and leads to higher costs. With many security units understaffed and underfunded, alternative cybersecurity strategies are essential [1]. The growing infrastructure, including mobiles, cloud, and IoT devices, means more threats and alerts. To counteract this, security teams must be equipped with the latest threat intelligence knowledge, including context, methods, indications, and actionable advice that informs about potential cyber threats [102]. Gartner [44] defines this as essential information that helps understand current or emerging threats to IT or information assets. An organization's incident response approach addresses cyberattacks through specific policies and procedures aimed at the containment of, as well as recovery from cyberattacks. Repetitive and tedious tasks often frustrate SOC staff, making it challenging to retain skilled members familiar with organizational systems and processes [101]. Using threat intelligence platforms to compile, aggregate, and organize threat data from multiple sources saves analysts time by providing up-to-date information on known threats and facilitating the communication of this intelligence within the organization and with external stakeholders.

In addition, **Security Orchestration, Automation, and Response (SOAR)** tools are gaining traction as essential components in security strategies, given the escalating security demands. SOAR technology enables the collection and utilization of data from security operations, defining and prioritizing incident response activities that blend human and machine capabilities for incident analysis and triage [84]. SOAR not only centralizes threat visibility but also automates routine tasks and, therefore, supports and scales the capabilities of human analysts [12]. Comparing SOAR with **Security Information and Event Management (SIEM)**, each has unique capabilities that, when combined, provide a comprehensive approach to incident response [93]. While SIEM systems are adept at managing and analyzing vast quantities of security data to identify threats, they are not as effective in handling the subsequent range of operations required for incident response. SOAR steps in with its advanced automation capabilities and the potential to reduce the need for human intervention in certain incident processes. By integrating SIEM and SOAR, organizations can optimize their security efforts. SIEMs handle the heavy lifting of data ingestion and alert generation, while SOAR systems facilitate the incident response process by automating and orchestrating routine tasks [12].

Academic research in the incident response domain is limited but growing, with studies exploring various challenges and proposing solutions. Shaked et al. [92] address the crucial role of incident response in organizational cybersecurity. They critique the lack of a disciplined approach in current playbook design and representation, which may affect their effectiveness. The authors propose a formal, model-based design approach for cybersecurity incident response playbooks, and introduce a tool prototype. Furthermore, Gurabi et al. derived requirements for the transition to structured security playbooks and their integration with other security tools for incident response, reporting, and automation. They extended the effort by developing a framework for a tool-assisted incident response based on playbooks [4]. Additionally, Schlette et al. [85] conduct an in-depth investigation into incident response playbooks. The research reveals ambiguities in how playbooks are defined and used by practitioners, finding that playbooks often cannot be applied directly across different organizations due to individualized definitions and varying focuses on incident response areas. In the context of response automation, Nespoli et al. [68] discuss the concept of countermeasures and the need for standardized security automation, highlighting issues like scalability and knowledge management. The authors of [75] glean insights from incident response analysts, emphasizing the importance of stakeholder engagement and maintenance concerns in automation, while warning against over-automation that could complicate tasks. Andrade et al. [10] review the literature on security incident processes, while [94] analyzes standards and decision-making gaps in incident analysis. Regarding incident-related models, various approaches are proposed, such as an automatic decision-making model [106], a deep learning-based event classification for SOCs [47], and a data mining approach to predict cyberattacks [50]. Additionally, network-focused research includes the development of systems like the Network Entity Reputation Database System [13] for characterizing network entities and predicting malicious activities. Moreover, Husák and Čermák [51] explore the use of recommendation systems in the context of incident handling and response. Their investigation revolves around the implementation, challenges, and various applications in this field. They particularly emphasize the role of recommender systems in identifying and prioritizing incidents, suggesting appropriate mitigation strategies, and offering real-time insights into evolving incident trends. Further, in [54], authors review automatic incident response solutions, categorizing them using the MITRE D3FEND framework and comparing academic approaches with commercial solutions.

The research contributions reflect a thorough effort to improve incident response through a combination of standardization, automation, and predictive analytics. In the following, we delve into three different elements of incident response in our work. First, we start with the cybersecurity playbook management acknowledging its importance in the current cybersecurity landscape. Subsequently, we examine the role of incident similarity in providing recommendations and enhancing the support of human operators. Lastly, we explore the automation of incident response.

6.1 Cybersecurity Playbook Management

The formulation, management, and sharing of cybersecurity playbooks are necessary for effective incident response and governance in an increasingly complex threat environment. Within the SAPPAN project, a novel vocabulary and process have been formulated to model response and recovery steps for incident response workflow documentation, leveraging semantic Web technologies. This methodology emphasizes the capturing, management, and sharing of knowledge to provide recommendations for manual operators or the automation of actions that reduce human intervention. The methodology and schematics of playbooks provide most of the functionalities that are also provided by the OASIS specification **Collaborative Automated Course of Action Operations (CACAO)** for Cybersecurity [79]. However, the SAPPAN vocabulary follows a more flexible approach in methodology and structure. This flexibility includes modeling confidentiality levels for single resources, which facilitates refined access control, data sanitization processes, and the formulation and dissemination of shareable playbooks. In the development of our vocabulary, we employ established semantic Web standards such as **Resource Description Framework (RDF)**, RDF Schema, and Web Ontology Language 2. It potentially provides the opportunity for easy integration of a knowledge base that follows a similar specification.

Further, SAPPAN introduces a semantic Web-based knowledge-capturing approach for playbook creation, storage, management, conversion, sanitization, and sharing [3]. It includes a tool based on **Semantic MediaWiki (SMW)**, which integrates semantic Web technologies into a MediaWiki knowledge base. As discussed in [69], it is necessary to sanitize the playbooks before sharing. Therefore, our process consists of a playbook sanitizer module, automating the extraction of public or shareable playbooks from confidential response and recovery processes. This proof-of-concept component eliminates manually labeled sensitive data based on the **Traffic Light Protocol (TLP)**. It subsequently generates a version of the playbook that can be disseminated to targeted organizations, specific departments, or security operation levels. Additionally, the system incorporates a playbook converter module. This feature is configured to import playbooks in SAPPAN or CACAO format directly into the management system and to export playbooks to those formats. Also, for sharing the playbooks, a playbook subscriber module enables searching and importing playbooks from the MISP platform and facilitates their distribution. Lastly, the steps within each playbook are schematically modeled and archived within the playbook management system. These steps are visually represented through **Business Process Model and Notation (BPMN)** diagrams, which can be interactively navigated and amended. In the course of the work, a number of response and recovery playbooks for malicious domain detection have been proposed and developed. This also includes generalization and sharing of the playbooks for adoption in other organizations.

6.2 Incident Similarity for Recommendation and Support of Human Operators

When handling a security incident, the goal of a SOC operator is generally to label the incident as indicative of truly malicious behavior (a true positive) or not (a false positive). If the incident represents a genuine attack, response actions are then initiated. In a naive attempt to use ML methods to assist in incident handling, one may try to train a classifier on historical data of labeled incidents and use its predictions to handle future incidents automatically.

However, labeling security incidents is a labor-intensive procedure, and the largest SOCs can handle no more than a few dozen incidents every day. On the other hand, the underlying features of a security incident are complex and heterogeneous: command lines, names of executables, file paths, file binaries, and so on. Therefore, in any realistic scenario, the size of the feature space dwarfs the number of labeled data points, making supervised ML models susceptible to overfitting. This situation is then complicated by the fact that labeling is an inexact procedure performed by security analysts of varying levels of expertise, and thus obtaining ground truth labels is a more inexact science in comparison to other domains such as image recognition.

It is then natural to use unsupervised ML models to assist with incident response. Perhaps the best-known unsupervised ML framework used in cybersecurity is anomaly detection (see, for example, [6, 22, 39]). However, anomaly detection methods rely heavily on having a large sample of clean data from which one can extract a probability distribution corresponding to non-anomalous behavior. Due to our extremely high-dimensional feature space, heterogeneity of the data, and the lingering uncertainty that no sufficiently large sample of incident data can be truly devoid of malicious behavior, modeling the underlying probability distribution of non-malicious behavior appears intractable.

Our ML framework of choice is instead a *similarity model*. Instead of making absolute judgments about individual incidents, we opt to train a model, which makes relative judgments in recognizing when two given incidents are similar to one another. To a security expert handling an incident, this can provide critical context: Given an active incident, one can query the model to discover how previous highly similar incidents were handled and act accordingly. This can increase the overall efficiency of SOC operations.

As part of the SAPPAN framework, we have constructed an incident similarity model whose training and inference algorithms function as follows. First, an incident vectorizer is trained on the past X number of days of incident data, obtained across all client organizations. The vectorized incidents used for training are then stored as part of the model. During inference, an end user, such as a security analyst, inputs a query incident from a given organization into the model, which is then vectorized by the trained vectorizer. Its cosine similarity with all incidents from the same organization in the training data is computed, and the most similar N incidents are returned to the user. Model validation is performed by a team of data scientists and security experts, meticulously going through a number of sample query incidents to verify that the corresponding similarity scores match expectations. To satisfy end-user demands, training and inference are both done rapidly with low-complexity algorithms, which take advantage heavily of the sparsity inherent in incident data.

To preserve data privacy with respect to individual organizations, end users, who may also be security experts employed at the given organization, are only shown similar incidents from their own organization during model inference. On the other hand, the vectorizer is trained on all data from all client organizations, allowing the vectorizer to properly understand individual features in a global context. Another approach would be to train individual models for each client organization; however, this leaves smaller organizations without sufficient data for training their own model without a way to compute meaningful similarity scores for incidents. In this way,

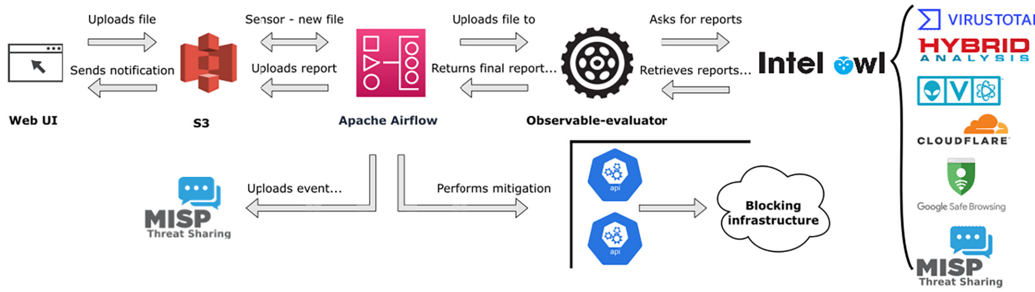

Fig. 4. Malware pipeline consisting of several microservices and tools orchestrated by Apache Airflow [97].

the incident similarity model demonstrates that sharing data across organizations in a way that model outputs still preserve client privacy can benefit SOC operations greatly.

6.3 Incident Response Automation

Responding quickly to an incident is one of the main objectives of security teams to prevent more severe impacts. However, a large amount of analysis is still done manually, and analysts often find themselves repeating the same procedures over and over again, just on different data, which prolongs incident response. An example of such a repetitive procedure is the analysis of suspicious software, where it is necessary to extract its identifiers (e.g., hashes, strings, and IP addresses) and verify in the threat intelligence database whether it is malware. Subsequently, a report should be generated, and appropriate mitigation steps should be taken if it is identified as malware. To demonstrate and validate how all these steps can be automated, we designed and implemented a malware analysis workflow [97], shown schematically in Figure 4.

The system consists of several components, as depicted in the architecture overview given in Figure 4. Here, we provide a short description of the key components:

— *WebUI*: WebUI is a simple React form application. It allows users to upload malware samples for evaluation and to read finished reports.
— *S3*: We use S3-compatible object storage MinIO to store malware samples and reports. The main advantages of MinIO are that it is lightweight, simple to interact with, and has official Docker and Apache Airflow support.
— *Apache Airflow*: Apache Airflow is an open source Python-based workflow and orchestration platform. It contains the orchestration logic of our process, periodically checks new uploads in the MinIO bucket, and sends malware samples to the Observable Evaluator. Furthermore, it obtains analysis results and triggers necessary reporting and mitigation procedures.
— *Observable Evaluator*: Our custom tool that is used for malware analysis. It sends samples to IntelOwl and collects analysis results. The main idea behind this tool is that IntelOwl simply uploads samples to third-party tools and returns their raw results. It performs neither “recursive analysis” (meaning analysis of observables, e.g., IP addresses, in malware sample) nor analysis evaluation (verdict or score).
— *Mocked Blocking Infrastructure*: To demonstrate the mitigation capabilities of the process, we are using a custom mocked blocking infrastructure closely based on the real-world infrastructure. It consists of an IP-blocking API and a domain-blocking API.

The aim of the proposed workflow and implemented prototype is not to undergo a deep analysis of malware behavior but rather to quickly evaluate a suspicious sample and automate the initial response. Moreover, we wanted to test the automation of the whole process, including the mitigation steps and discovering potentially infected machines in the organization’s network.

7 Task: Sharing

Transitioning from local attack detection and incident response to a collaborative perspective requires integrating some form of information sharing. In the following, we introduce some technologies relevant to sharing CTI and introduce sharing scenarios related to collaborative detection and response. We start by briefly mentioning some relevant formats, platforms, and sensitivity signaling approaches to provide a high-level overview of the sharing domain. The breadth of what can become cybersecurity-relevant data and the level of details render it very difficult to introduce a universal overarching standard that would capture all of them. Although there are frameworks that aspire to do so, such as STIX [78], we witness many other formats being used in production, e.g., due to lower complexity. CybOX [76] is used for describing cyber observables and cyber threats across various categories and serves as one of the building blocks for STIX. STIX structures information into key constructs, ensuring comprehensive and expressive threat sharing building on objects and relations. Common Event Expression [67] unifies and structures log and audit data, facilitating universal and machine-readable event records. OpenIOC [63] is a format for describing attacker activities and indicators, supporting extensibility. OpenC2 [77] standardizes the orchestration and automation of cyber defense and can be instantiated by the CACAO [79] format for playbooks. MISP [66] is a widely used platform for sharing intelligence on malware and cyber threats, expressed in JSON, encompassing objects, attributes, and taxonomies. As such, it provides its own data model and format which can be easily extended by adding new objects and taxonomies. MISP also implements distribution levels in its sharing platform; hence it is possible to define who receives the shared piece of information. Also, confidential information can be tagged by approaches such as TLP [43] or the **Information Exchange Policy (IEP)** framework [42].

When aiming to make sensitive data available to other parties, data sanitization can be considered as a way to prevent the leakage of sensitive information to the receiving parties. While anonymization (as a way to sanitize personal information) has been extensively studied in the past (e.g., [58, 62, 95, 110] for relational data and [52, 61, 109] for graph data), respective solutions for technical information have been more scarce, but format-specific approaches have, for example, been proposed for IP addresses [104] and packet traces [80]. Other approaches such as differential privacy [35, 36], **Secure Multi-Party Computation (SMPC)** [108], and **Homomorphic Encryption (HE)** [2] apply privacy-enhancing modifications to data processing tasks rather than the data itself. We want to emphasize that cybersecurity use cases may deal with sensitive information that is not necessarily of a personal nature. Using the term *privacy* in this context may hence conflict with privacy definitions that are strictly related to personal information. However, a clear distinction between these categories is complicated by the circumstance that it is not always clear whether a specific piece of technical information could ultimately be attributed to a person. Nevertheless, many approaches in the category of privacy-enhancing technologies can conceptually also be used to protect sensitive non-personal information. This is because in both cases, the ultimate goal is to avoid sharing information from which undesired consequences may arise. We thus focus on the protection of information for which sharing beyond a certain trust boundary (e.g., on an organizational level) could lead to undesired consequences.

In the following, we consider three different aspects of sharing in a cybersecurity setting. First, we consider sharing and federation for cyber threat detection, focusing on sharing federated learning updates to collaboratively train DGA detection classifiers. We then consider privacy and security risks associated with the training data used in the DGA detection use case, and briefly list the privacy-enhancing approaches that we have evaluated as part of our validation. Lastly, we outline our work done to enable sharing of cybersecurity playbooks.

7.1 Sharing and Federation for Cyber Threat Detection

The traditional approach of sharing for cyber threat detection is to share IoCs, such as a malware binary or an IP address of a C2 server. Our aim is to broaden the scope of sharing data between multiple organizations supporting privacy. We aim not to share the raw data but rather metadata such as the classification models. In such a case, the classification models become new IoCs, for example, represented by a neural network or a decision tree.

Once we are able to share models, we can establish various ML training schemes across organizations, such as federated learning. In federated learning, every party that is participating in the training of a global model first acquires the current model, which can also be a randomly initialized model at the beginning. Then, each party individually improves the current model using its private data and subsequently summarizes the changes to the model as a small, focused update. Every party distributes its model update and averages it together with the updates of all other participants. Using this federation step, every party can now apply the collaboratively computed update to the shared model to improve it. This is an iterative process, and multiple federation steps are performed until the global model reaches a maximum regarding classification performance on a certain validation set. All participating parties train a neural network classifier collaboratively by applying averaged model updates to a shared global model.

We investigate two different approaches for federated learning that differ in their type and amount of federation steps: federation after model convergence and federation after each training epoch. In the first case, either a randomly initialized or a pre-trained neural network classifier is distributed among all participating parties. This model is used as a starting point for all collaborating parties. The pre-trained global model can be trained based on a publicly available dataset (e.g., Alexa or Tranco top domain names for benign training samples in the DGA classification use case) so as not to leak sensitive information. All parties then continue the training of the initial classifier using their own private data. After each locally trained model converges (i.e., it reaches a maximum regarding classification performance on a validation set), all parties compute the updates in relation to the initial model. The updates equal the difference of the neural network classifiers' weights to the initial model. Subsequently, all updates are merged by averaging and applied to the initial model, which yields the final global model.

The second case, federation after each training epoch, is similar to the previous one but differs in the type and the number of federation steps. Instead of training the initial model locally up to its convergence, each party shares the computed update after each training epoch. Then, similarly, all updates are merged by averaging and applied to the initial model, which yields the global model for the next training epoch. All parties then continue the training of the updated global model for another epoch. This process is continued until the global model converges. The chosen approach for federated learning influences the global model's classification performance and the level of privacy provided.

The federated learning scenario is facilitated by the MISP platform with all the implications. As such, the sharing parties can define distribution levels such as sharing within the local MISP instance or sharing synchronized with other MISP instances (if there are partners willing to share their models and capable of processing the data model proposed for sharing ML models). The shared data must include the following attributes as a minimum:

- Classification Model: The actual classification model contains zero-weights for architecture description but the initial model must be randomly initialized in a global way.
- Model Formats: ONNX, HDF5, XML, numpy array (for model updates).
- Use Case: DGA classifier global init, DGA classifier update, DGA classifier final.
- Unique Training Identifier: An identifier used to synchronize the training procedure, e.g., a hash of the previous round.

The latest shared model is, in fact, an IoC and the organizations involved in the sharing can apply it in their cybersecurity analytic frameworks to discover suspicious events, such as the detection of AGDs in DNS **Non-eXistent (NX)** domain replies.

7.2 Cybersecurity Data Sanitization for Privacy- and Security-Aware Sharing

Whenever data are shared, the question of potentially negative consequences in doing so arises. Especially in the context of cybersecurity, concerns of unintentionally disclosing information that may benefit malicious activities have to be addressed. In the context of DGA detection, it is to be noted that only the benign samples of the training dataset (the collected NX domains) are considered to be sensitive, as malicious samples are available as OSINT. The benign NX domains, however, can be considered to be both security and privacy critical. From the security perspective, NX domains may leak information about software used in the network, potentially revealing the use of outdated versions with known security issues. When containing user-queried domains, the set of NX domains can further provide information about browsing behavior in the network. Such information may further provide information about the organization, in which the NX domains have been collected. This combination of known vulnerability and identifiability of the respective target poses a security risk. Another security concern is introduced by misconfigured devices, which try to connect to services that are no longer online, resulting in a NX domain request. In such a case, an adversary may take advantage of this situation and register the respective domain to mimic the respective service.

As such, the benign training data samples in the DGA detection use case should be protected when considering a scenario that involves sharing. Beyond addressing the above mentioned security concerns, the samples may also have to be protected to avoid the leakage of personal information, or to comply with organization guidelines and legal obligations such as data protection regulations or customer contracts.

Even though federated learning allows to benefit from diverse training data sources, it requires each contributing party to participate in the training process. Organizations with a lack of potent hardware may, hence, become a bottleneck in the training process. But also aside from requirements for participating in the training process, additional measures may need to be taken to address privacy concerns surrounding training data extraction from model updates. While respective concerns can be addressed by secure aggregation of model updates [17, 40] or the application of local differential privacy during training [98, 103], these approaches commonly take a toll on performance or model utility [59]. We thus also evaluated settings beyond federated learning. Specifically, we considered two approaches for training data sanitization as well as different cryptographic schemes for privacy-preserving **Classification-as-a-Service (CaaS)**. The sanitization of training data allows to create and utilize larger and more diverse training datasets from multiple sources without the requirement that every data provider actively needs to participate in the training process. This also allows for the asynchronous collection of training data, and provides the receiving organization more flexibility in how the data can be used, for example, in regard to the specific type of ML model that is trained on it. While sanitized training data can also be used to hide sensitive information in both a **Machine-Learning-as-a-Service (MLaaS)** and a CaaS setting, the evaluated cryptographic schemes provide strong formal guarantees, which are commonly more difficult to obtain for sanitization approaches. Specifically, we have considered the following approaches:

(1) We transferred a similarity-preserving encoding technique from the area of privacy-preserving record linkage to the DGA detection scenario for the purpose of training data sanitization [70, 71].
(2) We performed a privacy analysis of the FANCI feature extraction approach [90] to evaluate, whether the domains can be reconstructed based on their respective extracted feature vectors, and hence whether the FANCI feature extractor may be used as a sanitization approach in the DGA detection scenario [49].
(3) We evaluated existing frameworks that implement different SMPC and HE schemes as cryptographic approaches in the CaaS setting to protect samples during inference [32].

7.3 Cybersecurity Playbook Sharing

The academic and practical implications of sharing cybersecurity playbooks hold significant promise for the evolution of CTI ecosystems. Collaborative engagement in capturing and disseminating cybersecurity playbooks and their steps is important for the automation of response and recovery procedures [72]. In an effort to improve CTI exchange and facilitate collaboration, we essayed a requirement analysis [107] of the sharing platforms MISP, STIX, and OpenCTI. Further, we investigated the MISP architecture in more detail as the selected sharing platform. This investigation contains a systematic survey of MISP objects, tags, galaxies, attributes, and types, with the objective of developing a coherent data model for cybersecurity playbook sharing. The initiative also aims to create a sharing interface integrated within MISP's existing framework. In a methodological approach, we mapped the playbook along with its associated metadata to the existing MISP data model. This mapping exercise served as a critical point in identifying synergies between different playbook data structures and MISP's modular architecture. We implemented the MISP CACAO object, incorporating it into the official MISP repository. This effort not only extends the functionality of the MISP platform but also validates the feasibility of the newly developed data model and sharing interface.

8 Validation

Our validation addresses multiple facets of the SAPPAN architecture and the presented tasks. We evaluate improvements from a technical and quantifiable perspective where possible. We complement the technical evaluation with an opinion-based evaluation as expressed by a team of domain experts who assessed the SAPPAN framework on the specific use cases. In the following, we summarize our results and key findings per task (local attack detection, incident response, and sharing), and provide pointers to more detailed descriptions where available.

8.1 Local Attack Detection

8.1.1 Malicious Domains Detection. Our work on the detection of malicious domains focused on two use cases. First, we summarize our findings on phishing detection using CT logs, followed by our results on DGA detection.

Phishing Detection via CT Logs. In validating our proposed methods for local attack detection, our initial focus was on detecting malicious domains, specifically through phishing classification in CT logs. Detailed results of this evaluation can be found in [31]. The following text summarizes the main results. Our data processing pipeline is designed to efficiently collect, standardize, integrate, refine, and annotate data from various sources. This flexibility allows our methodology to handle complex datasets of different structures and origins effectively. Using this pipeline, we have compared the developed classifiers with those described in existing literature. Regarding computational efficiency, we benchmarked our system's performance by processing a week's consolidated CT logs, completing the task in approximately one day. To illustrate, a single pipeline operation (which integrates four distinct meta classifiers) was executed across 24 cores of an Intel Xeon Platinum 8160 processor running at 2.1 GHz, which required about 55 GB of RAM. Consequently, we anticipate that our framework is well-prepared to quickly assimilate and process the significant number of certificates added to CT logs in real time. Additionally, our design emphasizes scalability to ensure optimal performance as computational capacities are increased.

DGA Detection. In addition to the analysis of CT logs, we also focused on evaluating DGA classifiers in the real world as part of the methods for local attack detection. We trained 20 classifiers to classify an unfiltered 1-month recording. A detailed description of the results can be found in our paper [34], and the following text summarizes the main results. In our month-long observation, we thoroughly sifted the data against DGArchive [81], revealing 13,870 unique domains created by six separate DGAs. These domains are readily identifiable through rudimentary blocklisting methods. Remarkably, 17 of the 20 classifiers we used could accurately identify all known harmful domains, resulting in an average **True-Positive Rate (TPR)** of 99.997%. Subsequent analysis of the residual samples, flagged as positive by our classifiers, was undertaken to unearth novel DGAs. By implementing a semi-automated strategy, we scrutinized these marked samples to detect previously unknown AGDs. We used a comprehensive clustering approach that included domain length, constituent characters, top-level domains, query frequency, temporal occurrence span, Shannon entropy, and the presence of English words to classify these domains. Leveraging the combined power of DGArchive, domain expertise, and thorough manual examination, we sorted 5,833 enigmatic domains into eight separate clusters. Interestingly, six of these clusters likely correspond to unidentified DGAs, one matches with an elusive seed of the Bamital DGA, and another aligns with the signature of the Conficker DGA. Notably, the Conficker samples generated earlier than their expected operational timeframe explain their absence from the historical DGArchive records.

As part of our focus on malicious domain detection, we further explored the possibilities of interpretable visualization of deep neural networks for DGA detection. We used seven cybersecurity experts (CSIRT team members, cybersecurity data analysts, and visualization experts) for initial testing. The technical report [14] provides a detailed methodology and further evaluation description. The initial findings indicate that the suggested visualizations could lead to a more consistent trust in users' decisions and find value in the visualizations. In behavior-related tasks, participants develop a more intricate understanding of the model and frequently cite the analysis visualizations as having a substantial influence on their decisions. However, those utilizing the visualization tool required a significantly longer time to finish any task and needed an extended training period compared to the control group, likely due to the more intricate nature of the interface overall. Regarding expertise, the current data imply that beginners exhibit a higher average confidence in their decisions and explanations. Beginners in visualization also report a higher degree of usefulness than their expert counterparts, while the difference in reported usefulness for ML beginners and experts is less noticeable.

8.1.2 Anomalous Behavior Detection. In addition to detecting malicious domains, we focused on anomalous behavior detection, primarily on application profiling and process mining. In the case of fingerprinting based on DNS traffic, we validated the capabilities of the proposed approach on both simulated data and real data [91]. In our empirical analysis, Windows 10 was detected on roughly 11% of all hosts that initiated at least one DNS query. Among hosts where no particular application was identified, almost half (48%) sent less than 100 DNS packets in total. It is crucial to note that our datasets included a variety of devices, such as desktops, laptops, and mobile phones, but our analysis rules were specifically designed for Windows 10. Interestingly, our investigation revealed only a handful of fingerprints linked to crypto-mining activities. This scarcity could be due to the decreased presence of such applications during the observation period or potential variances in application behavior between our experimental environment and the broader dataset. Notably, our rule creation primarily resulted in a single label for Nicehash and three labels for Easyminer. Applications with more detections typically displayed a more comprehensive array of unique domain queries, thus providing a more diverse set of labels for analysis. Nevertheless, our tool exhibited notable effectiveness in identifying numerous unique fingerprints within the dataset.

To evaluate our tool for anomaly detection on application behavior using process mining, we used a simulated dataset consisting of *RED-benign* captured over 51 days on a simulated environment without any attack data and *RED-attack* captured during 1 day red-teaming experiment [48]. Our validation began by using the test dataset as input for our tool, in conjunction with the generated models, to ascertain the accurate depiction of benign application behaviors on the hosts. Following this, we introduced the attack dataset into the system to assess whether the trace checker would highlight increased errors indicative of anomalous activities. The results are presented in Table 3. Upon examining the test dataset, our tool demonstrated a precision of 97.22%. The discrepancies mainly originated from instances where no starting point was identified. This can be attributed to specific processes being operational before the initiation of our data capture. On the other hand, with the attack dataset, a higher rate of discrepancies was noted, with only 71.15% of events conforming to the established model. Of the remaining 28.85%, 6.59% were due to missing start points, which can be classified as false positives, as previously explained. Most of the discrepancies resulted from unexpected successors, suggesting that the subsequent events occurred earlier than expected within the trace or introduced entirely new sequences.

8.2 Incident Response

8.2.1 Playbook Management. The playbook management tool that allows a user to create, modify, and manage playbook steps and their relations was rated by the surveyed experts as very useful to help not only to create but also to understand the playbook steps via the BPMN graph representation. The SAPPAN playbook management

Table 3. Errors for the Test and Attack Datasets Compared to the Benign Model, Adapted from [48]

Tool output	**RED-test vs. RED-train**	**RED-attack vs. RED-benign**
true	97.22%	71.15%
true, but insertions	0%	4.12%
multiple fingerprints not part of the fingerprint matrix	0%	6.32%
successor is not correlated to the fingerprint	0%	10.16%
no startpoint found	2.78%	6.59%
endpoint not valid	0%	0.55%
startpoint not in the fingerprint matrix	0%	0.27%
endpoint not in the fingerprint-matrix	0%	0.82%

toolkit, incorporating SMW, provides functionality crucial for sanitizing, converting, presenting, and sharing playbooks. This involves a simple playbook sanitizer module that automates the extraction of public or shareable playbooks from confidential response and recovery processes, effectively removing labeled sensitive information. Complementary to this, the playbook converter and subscriber modules facilitate the import and export of playbooks in various formats and their distribution within the MISP platform. The playbook presentation module represents the response and recovery steps into BPMN graphs for interactive management. In our opinion-based evaluation, all surveyed experts rated the improved understanding of playbook steps and recommendations between 7 and 9 on a scale from 0 (does not help) to 10 (nothing better can be done), with an average score of 8.2.

8.2.2 Incident Similarity. The domain experts rated the ability to gather similar incidents and how they were handled as very useful. On average, they rated it between 7 and 8 on a scale of 0–10 of usefulness. They noted that the incident similarity improves the context knowledge of an incident handler and that this context knowledge, based on similar incidents handled in the past, can be used to reveal false-positive cases and support decisions on what should be the next steps when handling the incident based on the steps taken to deal with similar incidents.

In addition to the subjective validation of the incident similarity model as described above, we also validated the model quantitatively by computing the empirical distribution of similarity scores and confirming that it matched the security experts' expectations. To conduct this experiment, we sampled 5,000 similarity scores at random from 30 days worth of incident data, and plotted the resulting distributions in Figure 5.

The left plot in Figure 5 shows the distribution of all 5,000 randomly sampled similarity scores. Clearly, the distribution is heavily skewed towards 0. This is as expected, as two randomly chosen incidents will generally exhibit no meaningful similarity. The presence of a small, but noticeably non-zero, proportion of similarity scores, which are in the 0.1–0.2 range is explained by the presence of, for example, executables such as `cmd.exe` whose presence is ubiquitous in the incident data but signify little in terms of actual similarity.

In the right plot in Figure 5, we restrict our distribution to those similarity scores, which were at least 0.7. As customers, partners, and security experts are mostly interested in seeing pairs of incidents, which the model deems *are* similar, this represents a sample of model output, which would actually be shown to end users. As one can see from the plot, the distribution is concentrated close to 1.0, meaning the model is deeming a large proportion of incidents to be nearly identical in nature. Identifying such swaths of extremely high similarity scores enables mass processing of and response to incidents. Lastly, we note the spike in the distribution around 0.88. This was caused by two large collections of incidents that were validated by experts to be highly similar to each other.

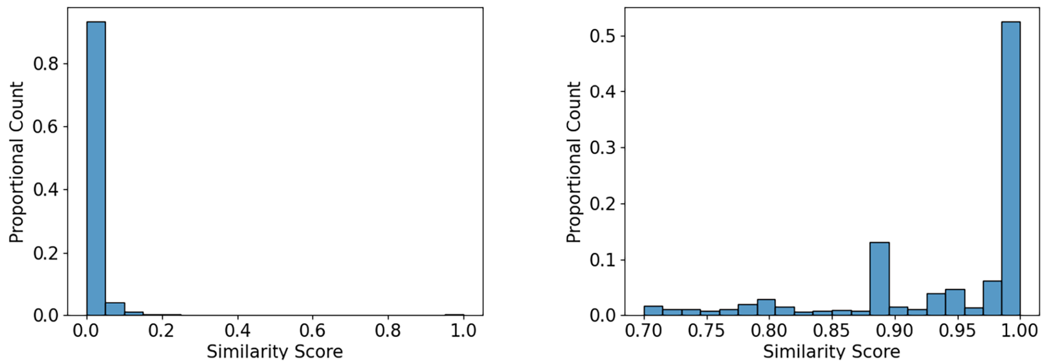

Fig. 5. Empirical distributions of incident similarity scores. On the left, the distribution of all similarity scores, and on the right, the distribution of those similarity scores which were at least 0.7.

8.2.3 Incident Response Automation. We have measured the malware analysis workflow to validate the benefits of incident response automation. We asked four professionals from a university CSIRT, responsible for daily operations and incident response, to go through the malware response process. Each analyst was given the same samples to process. Sample 1 was an MS Word document created by the test's authors to serve as a benign sample. Samples 2 and 3 were taken from the theZoo repository [105] and are actual malware samples. The goal was to decide whether a sample is malicious, as sometimes users report suspicious files that are benign. The analyst was asked to obtain network IoCs from the malicious samples and for each IoC to decide whether it is malicious or not. The next task was to block communication with malicious IoCs at the organization level. Finally, the analysts were asked to identify potentially compromised machines in the network by searching for the communication in network traffic records.

The results of the comparison of manual and automated malware analysis are in Table 4. The time measurement started with downloading the sample from the ticketing system, and the times in the table correspond to the times when the analyst finished the given task. In addition to time, we discussed the process with analysts and collected their feedback. They acknowledged that using the malware evaluator can reduce possible mistakes as a human can overlook or forget an IoC. They also mentioned as an advantage that they could upload the suspicious file and wait for the analysis results while doing other work instead of the manual analysis requiring their full attention the whole time. The validation also showed that the main challenge of automated processing is the availability of high-quality open source intelligence data, as several regular services were marked as malicious during the testing.

8.3 Sharing

8.3.1 Federated Learning. The sharing of ML models enables a collaborative approach for training without exchanging the data itself. In principle, this allows to utilize data that cannot be centralized (e.g., due to legal reasons). We demonstrated feasibility of federated learning in the DGA detection use case by training classifiers locally on data of a single institution and comparing this baseline to classifiers trained in federated fashion on data of four institutions [33].

For creating training and test datasets, we obtained malicious AGDs from one data source, and benign NX domains from four data sources. The malicious domains have been sampled from DGArchive [81], an open source intelligence feed that contained approximately 126 million unique domains generated by 95 different known DGA families at the time of sampling. The benign NX domains have been collected in four organizations (two universities, one association of universities, and one company). Using these data sources, we constructed four training and four test datasets, one for each source of benign samples, respectively. Each training dataset

Table 4. Comparison of Manual and Automated Processing of Potentially Malicious Sample Files

<table>
<tr><th></th><th></th><th></th><th>VirusTotal</th><th>Hybrid analysis</th><th>IoCs collection</th><th>IoCs evaluation</th><th>IoCs blocking</th><th>IoCs traffic</th><th>Total time</th></tr>
<tr><td rowspan="6">A1</td><td rowspan="2">S1</td><td>M</td><td>00:28</td><td>01:17</td><td>08:55</td><td>x</td><td>x</td><td>x</td><td>08:55</td></tr>
<tr><td>A</td><td colspan="2">00:25</td><td>01:29</td><td>x</td><td>x</td><td>x</td><td>01:29</td></tr>
<tr><td rowspan="2">S2</td><td>M</td><td>00:27</td><td>01:18</td><td>03:52</td><td>06:20</td><td>07:04</td><td>08:32</td><td>08:32</td></tr>
<tr><td>A</td><td colspan="2">00:31</td><td>02:02</td><td>02:02</td><td>02:02</td><td>02:02</td><td>02:02</td></tr>
<tr><td rowspan="2">S3</td><td>M</td><td>00:24</td><td>00:59</td><td>02:40</td><td>03:45</td><td>05:01</td><td>06:10</td><td>06:10</td></tr>
<tr><td>A</td><td colspan="2">00:15</td><td>02:17</td><td>02:17</td><td>02:17</td><td>02:17</td><td>02:17</td></tr>
<tr><td rowspan="6">A2</td><td rowspan="2">S1</td><td>M</td><td>00:20</td><td>01:07</td><td>01:42</td><td>x</td><td>x</td><td>x</td><td>01:42</td></tr>
<tr><td>A</td><td colspan="2">00:24</td><td>02:28</td><td>x</td><td>x</td><td>x</td><td>02:28</td></tr>
<tr><td rowspan="2">S2</td><td>M</td><td>00:21</td><td>00:46</td><td>02:12</td><td>03:04</td><td>x</td><td>x</td><td>03:04</td></tr>
<tr><td>A</td><td colspan="2">00:14</td><td>01:58</td><td>01:58</td><td>01:58</td><td>01:58</td><td>01:58</td></tr>
<tr><td rowspan="2">S3</td><td>M</td><td>00:15</td><td>01:02</td><td>01:53</td><td>02:20</td><td>04:01</td><td>05:24</td><td>05:24</td></tr>
<tr><td>A</td><td colspan="2">00:13</td><td>01:45</td><td>01:45</td><td>01:45</td><td>01:45</td><td>01:45</td></tr>
<tr><td rowspan="6">A3</td><td rowspan="2">S1</td><td>M</td><td>00:20</td><td>00:56</td><td>01:20</td><td>x</td><td>x</td><td>x</td><td>01:20</td></tr>
<tr><td>A</td><td colspan="2">00:20</td><td>01:24</td><td>x</td><td>x</td><td>x</td><td>01:24</td></tr>
<tr><td rowspan="2">S2</td><td>M</td><td>00:12</td><td>00:30</td><td>03:15</td><td>04:29</td><td>x</td><td>x</td><td>04:29</td></tr>
<tr><td>A</td><td colspan="2">00:16</td><td>02:43</td><td>02:43</td><td>02:43</td><td>02:43</td><td>02:43</td></tr>
<tr><td rowspan="2">S3</td><td>M</td><td>00:10</td><td>00:26</td><td>01:37</td><td>01:57</td><td>02:53</td><td>03:30</td><td>03:30</td></tr>
<tr><td>A</td><td colspan="2">00:12</td><td>02:52</td><td>02:52</td><td>02:52</td><td>02:52</td><td>02:52</td></tr>
<tr><td rowspan="6">A4</td><td rowspan="2">S1</td><td>M</td><td>00:54</td><td>02:21</td><td>02:41</td><td>x</td><td>x</td><td>x</td><td>02:41</td></tr>
<tr><td>A</td><td colspan="2">00:48</td><td>01:51</td><td>x</td><td>x</td><td>x</td><td>01:51</td></tr>
<tr><td rowspan="2">S2</td><td>M</td><td>00:16</td><td>01:24</td><td>02:24</td><td>04:31</td><td>x</td><td>x</td><td>04:31</td></tr>
<tr><td>A</td><td colspan="2">00:45</td><td>01:56</td><td>01:56</td><td>01:56</td><td>01:56</td><td>01:56</td></tr>
<tr><td rowspan="2">S3</td><td>M</td><td>00:15</td><td>01:27</td><td>02:08</td><td>03:40</td><td>04:38</td><td>05:38</td><td>05:38</td></tr>
<tr><td>A</td><td colspan="2">00:35</td><td>01:52</td><td>01:52</td><td>01:52</td><td>01:52</td><td>01:52</td></tr>
</table>

The bold column headers represent the different steps to process a potentially malicious file. A#, Analyst; A, automated analysis; M, manual analysis; S#, sample.

contained 446k samples, and each test dataset 222k samples. Each of these datasets had a 50/50 label distribution (malicious/benign) in regard to the DGA classification task.

To establish the baseline for local training, we performed five repetitions of training and evaluating a classifier for each combination of training and test dataset. In the following, we focus on the results of our ResNet-based classifiers developed in [34]. We took the average TPR and average FPR of these 80 evaluation runs to establish the local baseline.

For federated learning, we evaluated different setups. In the following, we present the result of training the ResNet-based classifier with a randomly initialized global model and federation after each epoch. We performed 5 repetitions for each test dataset and each of the 11 possible combinations of participating organizations. Again, we took the average TPR and FPR of these 220 evaluation runs.

The results are given in Table 5. A more detailed description of this evaluation as well as additional experiments can be found in [33].

Table 5. Improvements Gained by Federated Learning for the ResNet-Based Model Developed in [34], as Evaluated in [33]

Approach	TPR	FPR
Local	0.99844	0.01640
Federated	0.99935	0.00913

The comparison of local and federated schemes demonstrates a minor improvement in the TPR in favor of the federated approach. More importantly, the FPR was reduced by 44% in the case of federated learning. This is a significant improvement when translated into absolute numbers, as the number of non-existing resolutions can easily reach thousands per day in large organizations. The performance overhead caused by federated learning was represented in the majority by the longer time to train the classifier. The CPU and memory utilization was approximately the same for both approaches, but it took three times longer to finish the federated learning process (approximately 9 minutes) compared to local training (3 minutes). Communication and synchronization over the sharing platform caused the major overhead of federated learning.

8.3.2 Data Sanitization. Considering scenarios beyond federated learning, we have investigated two approaches for data sanitization (Bloom encodings and FANCI features) as well as the use of existing frameworks implementing different cryptographic approaches for CaaS. In the following, we briefly present our key findings evaluating these three approaches in the binary DGA detection use case (benign/malicious). The key benefits and shortcomings of the respective approaches are summarized in Table 7.

Training Data Sanitization via Bloom Filter Encodings. As the first approach for sanitizing training data, we considered a similarity-preserving encoding approach using Bloom filters. This encoding procedure has originally been used for the purpose of privacy-preserving record linkage [88], but provides properties that are conceptually suitable for ML use cases [70].

In regard to the general threat model, we considered a collaborative setting in which the encodings get shared with another party. We assume this party to be honest-but-curious with the goal of decoding the received encodings. In this context, two primary cases have to be distinguished: settings in which the encoding parameters remain secret, and settings in which they are accessible to the adversary. We have analyzed the threat associated with the former case more closely under consideration of a setting in which the encodings are sent to an honest-but-curious MLaaS provider for training and classification [71].

We evaluated the Bloom encoding approach using different values for the encoding parameters and different modes of encoding (basic, hardened through randomization [37, 89], and hardened through diffusion [11]) by training featureless CNNs on the respectively encoded versions of the same DGA dataset. The results are given in Table 6. For the basic and randomized encoding modes, a detailed description of the experiment can be found in [70]. The evaluation of the diffused encoding mode has not been published before, but uses the same dataset and experiment setup as in [70]. For the creation of the diffusion sets, the respective algorithm proposed in [11] has been used. The values of the basic encoding parameters have been adjusted in a privacy-driven fashion to suit the ML use case, prioritizing the protection of samples over model utility.

While we observe that the encoding approach has a notable negative impact on the TPR of the trained models, the impact on the FPR in absolute numbers is significantly smaller. As to be expected, applying hardening approaches further impacts the model's utility. Especially the use of diffusion greatly impacted both the TPR and FPR of the model (despite a relatively small diffusion set size of $t = 3$) in the case of 128 bit Bloom filters. The application of randomization had a smaller negative impact, providing an FPR close to the one of the basic Bloom encoding approach. Choosing shorter Bloom filter sizes (here 64 bit instead of 128 bit) increased the negative impact of the randomized approach on both TPR and FPR, but disproportionately affected the FPR.

Table 6. Comparison of the TPR and FPR for CNNs Trained on Differently Encoded Versions of the Same DGA Dataset (101,866 Samples with a 50/50 Label Distribution)

Basic encoding parameters	Hardening approach	TPR	FPR
Cleartext baseline	-	0.99545	0.00030
$l = 128, k = 2, q = 2$	None	0.94726	0.01343
$l = 128, k = 2, q = 2$	Randomization ($f = 0.4$)	0.87944	0.01701
$l = 128, k = 2, q = 2$	Diffusion ($t = 3$)	0.83848	0.05319
$l = 64, k = 2, q = 2$	None	0.90951	0.00754
$l = 64, k = 2, q = 2$	Randomization ($f = 0.4$)	0.84423	0.06670
$l = 64, k = 2, q = 2$	Diffusion ($t = 3$)	0.82438	0.05163

Parameters: l is the Bloom filter length, k the number of hash functions, q the q-gram length, f the randomization parameter of RAPPOR's randomized response step, and t the size of diffusion sets. The TPR and FPR of the cleartext baseline, basic encodings, and randomized encodings are based on the experiments carried out in [70].

Table 7. Key Benefits and Shortcomings of the Different Privacy-Enhancing Approaches Observed in the DGA Detection Use Case Based on [32, 49, 70, 71, 90]

Approach	Benefits	Shortcomings
Bloom encodings	—Provides many parameters to fine-tune the encodings for a desired privacy-utility tradeoff —Computationally efficient —Easily transferable to other use cases (as long as samples are roughly of similar length) —Can conceptually be used to share sanitized samples —Can be used in the MLaaS setting to hide the actual ML task from the MLaaS provider	—Selection of parameter values requires good understanding of the procedure —Privacy-driven selection of parameter values can have a notable negative impact on classification accuracy —Privacy guarantee not as strong as for cryptographic approaches —Sharing sanitized data for independent use requires receiving parties to know the encoding parameters to map other samples to the same space
FANCI features	—Very high utility —Computationally efficient —The feature extraction procedure does not seem to be efficiently reversible in general —Can conceptually be used to share sanitized samples	—Protection against other threats is unclear (e.g., membership inference, extraction of statistical information) —Changes to feature set require reconsideration of the reversibility property (as potentially required when applied to a different use case)
SMPC, HE	—No negative impact on the quality of classification results (if model is not notably simplified) —Strong guarantees for protecting samples in the CaaS setting —Allows to train on and classify unaltered samples without revealing them to collaborators	—Excessive communication cost (even with model simplifications) —Computationally intensive due to the complexity of cryptographic operations —Does conceptually not allow to share sanitized samples for independent use

For the diffusion approach, however, the shorter Bloom filters surprisingly had no notable impact. We expect that the model utility could be increased by using longer Bloom filters and more hash functions, but at the cost of potentially increased vulnerability to decoding attempts, e.g., via pattern mining attacks [23, 24, 100]. To evaluate the protection provided by the parameter values used above, we have performed a respective analysis considering basic and randomized Bloom encodings for DGA detection in the MLaaS setting [71]. This analysis did not find notable information leakage, and applied decoding attempts through pattern mining attacks remained unsuccessful.

Training Data Sanitization via FANCI Feature Extraction. In addition to obfuscating samples via Bloom encodings, we have also considered the privacy properties of feature vectors extracted via the FANCI feature extractor [49, 90]. As the FANCI feature extractor has originally been introduced to improve classification results in DGA detection, it is not expected to negatively impact utility when applied as a sanitization approach in this use case. The results of a respective utility evaluation can be found in [90]. The use of extracted features instead of cleartext samples, however, also provides some level of obfuscation. To evaluate the degree to which the FANCI feature extraction process is reversible, and hence whether a cleartext sample can be reconstructed from its extracted FANCI features, we have trained recurrent ML models on three different benign DGA datasets to learn this sample reconstruction task. A detailed description of our experiment can be found in [49]. We used the Damerau-Levenshtein distance metric [30] to quantify for each reconstruction how many operations (character substitution, character insertion, character deletion, transposition of adjacent characters) are at least required to construct the actual samples from the reconstruction output by the model. To account for differences in sample length, we additionally considered a normalized version of this distance, which divides the Damerau-Levenshtein distance by the length of the actual sample or the reconstruction, depending on which is longer. Considering the Damerau-Levenshtein distance and its normalized version for each combination of the three datasets (for the purpose of training and reconstruction), the most successful reconstruction attack still required an average of 13.66 operations (with a normalized value of 0.46), and in the least successful case an average of 72.94 operations (with a normalized value of 0.75) to retrieve a cleartext sample from the reconstructed one [49]. This indicates that reliably reversing the FANCI feature extraction process in general is not possible.

Nevertheless, the result does not guarantee that the use of FANCI features cannot leak any potentially sensitive information. For example, the protection against membership inference attacks on datasets of FANCI features and the extraction of statistical information, which may reveal sensitive information about the origin network, are not addressed by this evaluation. It should also be considered that changes to the set of extracted features, which might be required when a different use case is considered, may not provide the same properties in regard to general reversibility.

CaaS via SMPC/HE. Specifically for the CaaS setting in which a model owner trains a model on their own data and classifies samples of other parties as a service, we have evaluated different frameworks that implement cryptographic approaches, which allow to keep the samples hidden from the service provider. The detailed description of this evaluation can be found in [32]. While also the above mentioned sanitization approaches could be used in the CaaS setting by performing both training and inference on sanitized samples, the cryptographic approaches provide strong formal guarantees for protecting samples without sacrificing model utility. We have evaluated the performance of four existing frameworks, of which three implement SMPC approaches (PySyft [83], TF-Encrypted [28], and SecureQ8 [29]), and one implements a hybrid approach that combines SMPC and HE (MP2ML [16]). PySyft and TF-E implement 3-party protocols, while MP2ML implements a 2-party protocol. SecureQ8 provides both 2- and 3-party protocols.

We evaluated these frameworks with four state-of-the-art models (three deep learning models and one feature-based approach utilizing FANCI) on the same DGA dataset in the binary DGA classification task. The inference latency when applying the evaluated frameworks out of the box was prohibitive. Even though we achieved a reduction in inference latency by up to 95% and a reduction in communication overhead by up to 97% through model simplifications at an acceptable reduction in classification accuracy by less than 0.17%, the resulting inference latency per sample still showed to be too high to be applicable to real-world DGA detection scenarios [32]. While still over the desired inference latency of 405 to 430 µs, the evaluated 3-party protocols showed to be significantly more performant than the 2-party protocols, but it remains unclear how to integrate a third party into the CaaS setting in a meaningful way.

8.3.3 Playbook Sharing. Cybersecurity playbooks represent another specific type of data that is shared within the SAPPAN framework. We proposed and implemented a dedicated playbook-sharing object and rendered it publicly available through the official MISP repository. The object is fully compatible with the CACAO standard and is ready for any additional playbook formats and versions. The universality of the object itself is due to its straightforward design. The object consists of metadata describing the playbook properties and the playbook itself in the original format. The receiver is always capable of reading the metadata and can thus figure out the format of the playbook. This allows the receiver to identify, whether the playbook itself is in a desired format. Additionally, the MISP security playbook object can be tagged with additional context information such as MITRE (e.g., an attack pattern) to provide additional contextual information, or IEP/TLP to signalize the level of sensitivity.

In order to receive feedback from the domain experts, we asked how well the playbook sharing via MISP supports the response and recovery process in their organizations. The experts agreed that sharing the playbook in a standard format helps to receive and apply the playbooks. However, they noted that playbook sharing is primarily useful for similar organizations. If the environments of organizations are different, playbooks need to be adapted manually to the local needs to improve the response. The experts also mentioned that the CACAO playbook standard is relatively new and has not yet been adopted, as it was introduced in 2021. Additionally, the SAPPAN effort resulted in a joint publication with the CACAO committee [64]. This documents the initiative work carried out in the field of CTI sharing by the SAPPAN project, which not only introduced playbook sharing via MISP but has also made further recommendations for the enrichment of threat intelligence collaboration, particularly in the context of playbook sharing.

9 Discussion

In the following, we discuss the implications that each of our results have from three different perspectives per result. First, we provide a general discussion of the respective validation result. Secondly, we discuss the implications the result has in regard to the challenges and potentials of collaboration and automation as identified in Tables 1 and 2. And lastly, we align the result with components in the SAPPAN reference architecture as described in Section 4.

Detection of Malicious Domains and Anomalous Behavior. Local threat detection is the initial step in ensuring the security of the protected infrastructure and provides input for both incident response and collaboration through information sharing. Currently, there are a large number of different tools and techniques that can be used for this purpose. To demonstrate some of these techniques, we have chosen to detect malicious domains and anomalous behavior using ML techniques, which are now typical for modern threat detection. Evaluation of these techniques has shown their potential not only as stand-alone detection mechanisms but also as a suitable source of shared data when they provide sufficient accuracy.

The selected techniques are intended to show how the identified objectives can be approached and what potentials arise from them. In the case of DGA detection, we have proposed a tool that allows the use of different ML algorithms and compare them with each other, which has a potential for incident response (C.C5), where the analyst can not only use the result of the detection itself but also compare this result with other techniques to gain further information about the detection validity. Together with the proposed anomaly detection techniques, our methods address the potential of preemptive defense (C.P1), where ML allows us to detect novel kinds of attacks effectively. In addition to the detection mechanisms, we also focused on supporting the analysts in developing novel methods. We presented a tool for understandable interpretations of DGA classification, demonstrating how to increase confidence in ML model interpretation even in the face of a complex problem domain (C.P4). Consequently, increasing trust also plays an essential role in the possibilities of information sharing and further use.

Local detection mechanisms form the initial part of the SAPPAN reference architecture. Based on the outputs of these mechanisms, incidents are created and further managed through case management. At the same time, the outputs of local detections form the primary information shared through the sharing system, where local attack detection methods of other organizations may utilize it.

Incident Similarity. Our validation of the incident similarity-based methods has shown that incident similarity clustering is instrumental for the mass processing of similar incidents. According to our empirical experiments as well as according to our domain experts, it is a pivotal tool to automatically identify the FPs that fall within the cluster of incidents previously tagged as false positives within the context of similar organizations.

From this viewpoint, the incident-similarity clustering aligns effectively with the potential for faster processing and resolution of incidents (A.P2). The method utilizes the knowledge of human analysts based on their previous decisions on whether similar incidents were discerned as false positives. It serves as a bridge between human expertise and automated processes to reduce the workload on the human operators (A.P1). The clustering of similar incidents and their visualization also offers the opportunity to improve human–machine teaming (A.P4) by observing and analyzing the clusters of similar incidents across multiple organizations, thus increasing the visibility on a higher level to discover, e.g., attack campaigns targeting a specific sector.

The incident clustering functionality has enabled security experts to observe common activities across multiple customer organizations, which yielded information used to fix repetitive false-positive detections. As such the method is primarily beneficial for cybersecurity service providers or large organizations who deal with a high number of incidents. Within the SAPPAN reference architecture, its functionality is best aligned with the recommendation system component.

Incident Response Automation. The evaluation of the analysis workflow automating typical incident response steps related to malware infection showed the advantages of linking shared data and automated processing. Although analysts typically use shared data, their manual querying during analysis significantly delays the entire incident resolution. Our proposed and evaluated automated processing appropriately shows how this problem can be overcome and nicely demonstrates the potential of this approach for other typical analyses performed in incident response. The evaluation also showed that automation is very good at reducing potential errors by analysts who may miss essential information during manual analysis. In addition, it also shows that results are consistent across analysts, and that the analysis times are similar with only small differences.

The malware analysis workflow appropriately demonstrates the potential associated with automation in threat detection and response. Primarily, it shows the strength of workload reduction (A.P1), where it efficiently implements common steps associated with malware analysis, as the analyst can upload a suspicious file and attend to other incidents while the file is being processed and analyzed. This is also related to the overall speeding up of the incident response (A.P2) as the individual analysis steps do not need to be performed manually, but it is sufficient to confirm the resulting incident response actions depending on the analysis result. Using a standardized procedure and available tools also offers easy monitoring of the entire analysis process and possible evaluation of all performed steps (A.P3). A vital role for the proper functioning of the workflow is the shared data based on which the analyzed sample is evaluated. Its quality (C.C5) significantly influences the processing results. Within the workflow, it is possible to use several different data sources and balance them against each other, which can significantly refine the analysis results (A.C4) and allow the analyst to trust the results of the automated processing.

The malware analysis workflow is part of the automation engine and is directly linked to the case management and dashboard in the SAPPAN reference architecture. At the beginning of the incident resolution, the analyst uses the integrating dashboard to upload a suspicious file according to the recommended steps. Subsequently, all the steps of the analytical workflow are executed, and the analyst only confirms the suggested actions within case management (e.g., blocking communication with malware-related domains on the firewall). Although not directly stated in Figure 2, the sharing system plays a key role, from which the malware analysis workflow takes the CTI and relevant IoCs, which the analyst can further act upon as part of the incident response follow-up.

Collaboration and Sharing. The results obtained from federated learning implemented over the MISP platform underscored its efficacy in collaborative model training while preserving data privacy and reducing FPR by training on a larger and more diverse dataset. Additionally, the shared models become the new form of IoCs in addition to the traditional forms such as hashes and IP addresses.

From this viewpoint, the sharing of models aligns effectively with the potential for preemptive defense (C.P1). The shared models encapsulate the detection of attacks of higher complexity or stealthiness than those typically described by the conventional forms of IoCs and the sharing itself enables other organizations to prepare the detection even before the attack hits them. At the same time, federated learning over the popular sharing platform offers the opportunity to involve multiple organizations to train models over more diverse and balanced datasets (C.P2). Additionally, the federated learning scheme offers the possibility to share metadata rather than the sensitive data itself, thus partially addressing the challenge of dealing with sensitive information (C.C1).

The validation of federated learning within the SAPPAN architecture centered on the DGA classifier use case. However, both the training process and the sharing of models remain agnostic to the specific use case. The successful experimentation involving federated learning across three organizations using MISP demonstrated the viability of this approach. A minor concern may be that all partial models exchanged during the training epochs remained stored in MISP. On the one hand, this may serve well to analyze the training phase, but on the other hand, it pollutes the MISP storage. Therefore, it may be practical to remove these partial models which requires each participating organization to remove its respective models after the training phase.

Data Sanitization. Our evaluation of different privacy-enhancing approaches in the binary DGA detection use case showed the difficulties of making sensitive information shareable. Data sanitization approaches that aim to make the data itself shareable and provide the receiving parties more flexibility in how the data can be utilized face the challenge of difficult-to-quantify privacy guarantees (as indicated by privacy evaluation of the FANCI feature extraction approach) or require finding a tradeoff between data protection and model utility (as indicated by the evaluation of the Bloom encoding approach). While alternative approaches such as SMPC and HE can be used to target a specific data processing task rather than the data itself, the evaluation of different available frameworks implementing SMPC and HE approaches revealed that the high communication cost introduced by these approaches renders their use infeasible for DGA detection in real-world settings, even after models have been simplified to reduce communication cost. Further model simplifications are expected to negatively impact the accuracy of the model. We found the identification of a suitable tradeoff between protection of samples and utility to be a labor-intensive process. This has multiple reasons:

(1) It is not always clear what a suitable level of protection is. As universal protection is commonly not achievable, the question of realistic threat models arises. The evaluation against specific threat models, however, further limits the general applicability of a solution, as seemingly small changes to the threat model may drastically weaken the protection provided. Hence, careful consideration may be required when applying a privacy-enhancing technology to a new use case or setting.
(2) Utility can be measured in various different ways. In the case of our Bloom encoding experiments, the utility was primarily determined by the impact on classification metrics. For the evaluation of cryptographic protocols, however, the utility was primarily impacted by the overhead caused by the protocols. This overhead could be reduced through model simplifications, which in turn impacted the quality of classification results. The evaluation of utility may thus require to consider additional tradeoffs, e.g., between computational overhead and classification results.
(3) It is difficult to determine how much utility can reasonably be sacrificed for the protection of data. Especially in practice, the perspectives of different roles may collide. On the one hand, ML engineers and incident handlers may prioritize the utility of developed tools to reduce the number of false positives. On the other hand, privacy engineers and data protection officers may prioritize the protection of data to minimize the

chance of negative consequences arising from collaboration. Finding a suitable tradeoff can hence also require negotiation between different roles in an organization.

(4) The process of identifying a suitable tradeoff between utility and protection of data for a specific use case depends on the chosen privacy-enhancing approach. For the three approaches we evaluated (Bloom encodings, FANCI features, and cryptographic protocols), the starting conditions for identifying such a tradeoff differed. For using the FANCI feature extractor as a sanitization tool, a good utility was given, but the question of its protective properties arose. In contrast, the cryptographic protocols provided strong mathematical guarantees and did in theory not negatively impact classification results, but utility was limited by the introduced overhead. For the Bloom encodings, neither the level of protection nor the provided utility was fixed, but had to be tuned via parameter values. Nevertheless, the question of whether a different approach could provide a better tradeoff in a specific use case always remains.

In terms of the identified challenges and potentials of collaboration, we found the primary challenge to be the choice of a suitable tradeoff between the risk of leaking sensitive information (C.C1) and the quality of sanitized data (C.C5). The secondary challenges were the universality of the trained models (C.C3), the machine-interpretability of data (C.C2), and the confidence in automated decisions (A.C4). Model universality was affected by the evaluated privacy-enhancing approaches, as they either utilize a specific data domain (Bloom encodings or FANCI feature vectors) or require different parties to take part in a specific SMPC/HE protocol. In regard to machine-interpretable data and automated processing, the choice of a suitable privacy-enhancing approach for a specific setting as well as the choice of respective parameter values requires human consideration. Potential future advancements through standardization efforts (C.P3) may help to mitigate this challenge through standardized data sanitization procedures for common data formats and use cases, providing practitioners with guidance in the selection of privacy-enhancing approaches and their parameter values. Even though none of the privacy-preserving approaches discussed as part of this work provided a perfect balance between privacy and data utility, we are optimistic that future improvements in respective technologies can leverage collaborative data generation and learning (C.P2), which was also indicated by our comparison between DGA detection classifiers trained in the local and the federated setting. We further see advances in the field of privacy-enhancing technologies contributing to preemptive defense (C.P1) by allowing organizations to share relevant data with a wider range of recipients in case they observe a novel kind of attack.

In the context of the SAPPAN architecture, the evaluation of different privacy-enhancing measures in the DGA detection use case indicates that a general and use case independent implementation of the data sanitization and transformation component may not be possible. Nevertheless, and due to its importance for activities related to sharing, a modular use case specific approach could be taken to implement this component. As such, independent tools could be developed, each of which focuses on the sanitization of specific data formats in specific use cases. It should, however, be considered that a combination of differently sanitized versions of the same dataset can leak information that cannot be retrieved when observing each sanitized version independently. We hence recommend keeping an internal record of which data has been shared in which form with other parties.

Playbook Management and Sharing. Our evaluation of the effective management and sharing of playbooks draws similarities from previous discussions on privacy-enhancing measures in data sharing, and explores the complexities and importance of playbook sharing through the SAPPAN architecture. One of the project's significant contributions is the development of a dedicated playbook management tool, the integration of a sharing object into the official MISP repository, and the verification of the toolkit, characterized by its compatibility with the CACAO standard. Its universality arises from a straightforward design including metadata that describes playbook properties and the playbook in its original format. The approach to formulating playbooks arrives in a semantic Web-based knowledge-capturing methodology. The flexibility inherent in the SAPPAN vocabulary, which allows for modeling confidentiality levels of individual resources, is an advancement in sharing. It facilitates refined access control and data sanitization to enable the dissemination of shareable playbooks.

Machine-readable playbook creation, management, and sharing open a new avenue in collaboration and automation potential. They especially address the potentials of response standardization (C.P3) and increased quality via human–machine teaming (A.P3), and may further help to reduce human workload (A.P1) and enable faster processing and resolution of incidents (A.P2) [72]. However, several challenges emerge in the context of playbook sharing and management. Firstly, maintaining confidentiality by addressing sensitive information (C.C1) is crucial in playbook management and sharing. Another significant challenge is the adaptability of the data model (C.C4). Playbooks can reveal sensitive information that is, for instance, related to the operating devices, network topology, and possible vulnerabilities of the sharing party systems. The introduction of privacy-preserving measures, such as data sanitization, addresses this challenge but also introduces complexities in maintaining playbook effectiveness. The balance between sharing comprehensive playbooks and maintaining confidentiality is a challenge that can be addressed by a dynamic data model such as the SAPPAN vocabulary based on semantic Web standards which offers flexibility in defining resources and proposing a confidentiality level for each resource. Secondly, the challenge of machine-interpretability (C.C2) of shared information is addressed by the machine-readability of playbook formats. The efforts on more effective cybersecurity response automation is an interesting topic for future work in this regard. Thirdly, the universality of playbook models (C.C3) raises questions about the adaptability of playbooks across diverse incident scenarios. Compatibility with well-known and widely used standards such as CACAO can address this concern and hint at where simplified processes could enhance collaborative efforts. Lastly, dealing with different nuances of an incident (A.C3), particularly in the context of supporting workflow branching in playbooks, is supported by both the CACAO and the SAPPAN format. While SAPPAN has made a significant effort in playbook management and sharing, it navigates a complex landscape of challenges and potentials. The efforts in aligning with established standards, developing a flexible vocabulary, and integrating sanitization and sharing mechanisms mark a step forward in collaborative cybersecurity [5]. However, the continuous evolution of cyber threats and organizational environments necessitates ongoing refinement and adaptation of these approaches.

Within the context of the SAPPAN reference architecture, the assessment of playbook management and sharing emphasizes the necessity for a user interface component dedicated to the creation, management, representation, and sharing of playbooks via the dashboard interface. A data sanitization module is critical to filter out confidential details from the playbooks and their respective steps prior to their distribution as CTI. It is also critical to establish various sharing levels, tailored for distinct dissemination scenarios and for parties with varying degrees of trustworthiness. Moreover, playbooks in disparate formats should be converted within the data transformation module to facilitate integration and enhance interoperability. Concerning automation efforts, it is advisable to identify low-impact incidents and integrate the relevant executable source code, which synchronizes with the organization's existing systems, into the playbook procedures. While full automation of response playbooks presents considerable challenges, partial or semi-automation of routine with low-impact tasks is promising for practical application. Moreover, linking the playbook management tool with the case management system is beneficial, not only to assist human operators and enhance recommendation algorithms, but also to collect feedback and insight to refine playbooks for subsequent incident management.

10 Conclusion

In this work, we have taken a practical look at the potentials and challenges that collaboration and automation may bring to a multi-purpose sharing infrastructure in the cybersecurity domain. As such, we started by identifying respective key potentials and challenges. To additionally provide a more technical context, we have proposed the SAPPAN reference architecture as a general set of technical components and their interactions that allow the execution of common tasks for sharing-enabled incident response. We then presented our work on the tasks of local attack detection, incident response, and sharing. Based on our validation results, we discussed the implications for the previously identified key potentials and challenges and the SAPPAN reference architecture.

Our results showed that both collaboration and automation have the potential to strengthen detection, analysis, and response capabilities of an organization. The consideration of ML models as IoCs allows to establish preemptive defense measures that can deal with novel incidents. The use of more diverse training datasets, e.g., via federated learning, further allows for creating better classifiers, which can increase the confidence in automated detection and reduce human workload due to lower FPRs. Collaboration in the security domain, however, has to deal with the matter of sensitive information, ranging from training data over model updates to playbooks. In our evaluation of respective approaches for the DGA detection use case, we saw that none of the three evaluated approaches poses a perfect solution. Handling sensitive information thus still requires human consideration and is expected to negatively affect the quality of data or restrict the universality of the shared data. When considering sharing response procedures as machine-readable playbooks, it is crucial to utilize an adaptable and universally applicable data model. The data model must be sufficiently flexible to cover different nuances of an incident within one playbook. In this context, the sharing of such playbooks would benefit from standardized and universally recognized playbook formats. Respective efforts have recently been made with the CACAO playbook format. Apart from that, the sharing of playbooks may further contribute to the standardization of common analysis and response procedures. In that regard, also the automation of such procedures should be considered, as seen by our implementation of the automated malware analysis workflow. The automation of steps commonly carried out for analyzing a malware sample can benefit human operators by reducing human workload, providing faster processing of the respective steps, and providing a detailed log of the entire analysis process. This benefits human–machine teaming without notable risk of disruption. Further, human operators can be assisted by incident similarity clustering, allowing to connect experience from incidents handled in the past to current incidents. By keeping a human in the loop and using automation procedures primarily to complement the work of human operators by carrying out common analysis tasks and suggesting response actions (rather than carrying them out directly), the risk of disruption is kept low.

While our validation results had implications for a majority of the identified key potentials and challenges, some of them have been discussed to a lesser extent. To provide some interesting directions for future work, we briefly discuss them in the following. In our work, we primarily focused on automating analysis tasks rather than response actions, since the latter has a greater potential for disruption. It further is associated with a higher implementation effort, as it needs to interact with different systems in an organization [72]. Further, the exploration of visualization techniques is of great interest, as it can assist in the assessment of data received from a collaborator, especially for complex forms of data such as trained deep learning models. The use of visualization techniques should also be considered in the context of automation of workflows, as it could increase confidence in automated decision making and benefit human–machine teaming through better monitoring. While we discussed problems that arise from sharing information that is potentially sensitive in an ML setting, we have not focused on problems arising from malicious collaborators. In this regard, methods for detecting data poisoning attacks are of significant interest to evaluate the quality of received data. Lastly, we also want to point out the importance of sanitization procedures for playbooks. Since cybersecurity playbooks can provide rich information to guide human operators in their work, it comes at the cost of an equally rich data format. Efforts to make playbooks shareable thus strongly rely on suitable sanitization procedures to ensure that organization-specific information is removed from a playbook that is prepared for sharing.

Acknowledgments

We would like to thank all members of the SAPPAN consortium, the project's advisory board, and its end-user committee. Our specials thanks go to Gabriela Aumayr, Franziska Becker, Benjamin Heitmann, Tomas Jirsik, Alexey Kirichenko, Karl-Heinz Krempels, Martin Laštovička, Ulrike Meyer, Christoph Müller, Josef Niedermeier, Mischa Obrecht, and Robert Rapp who made this work possible.

References

[1] Md Sahrom Abu, Siti Rahayu Selamat, Aswami Ariffin, and Robiah Yusof. 2018. Cyber threat intelligence—Issue and challenges. *Indonesian Journal of Electrical Engineering and Computer Science* 10, 1 (2018), 371–379.

[2] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. 2018. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys* 51, 4, Article 79 (Jul. 2018), 35 pages. DOI: https://doi.org/10.1145/3214303

[3] Mehdi Akbari Gurabi, Avikarsha Mandal, Jan Popanda, Robert Rapp, and Stefan Decker. 2022. SASP: A semantic web-based approach for management of sharable cybersecurity playbooks. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, 1–8.

[4] Mehdi Akbari Gurabi, Lasse Nitz, Andrej Bregar, Jan Popanda, Christian Siemers, Roman Matzutt, and Avikarsha Mandal. 2024. Requirements for playbook-assisted cyber incident response, reporting and automation. *Digital Threats* 5, 3, Article 34 (Oct. 2024), 11 pages. DOI: https://doi.org/10.1145/3688810

[5] Mehdi Akbari Gurabi, Lasse Nitz, Charukeshi Mayuresh Joglekar, and Avikarsha Mandal. 2024. Strengthening cyber defence through cooperative development and shared expertise in incident response playbooks. *ERCIM NEWS* 139 (2024), 44–46.

[6] Montdher Alabadi and Yuksel Celik. 2020. Anomaly detection for cyber-security based on convolution neural network : A survey. In *Proceedings of 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 1–14. DOI: https://doi.org/10.1109/HORA49412.2020.9152899

[7] Bushra A. Alahmadi, Louise Axon, and Ivan Martinovic. 2022. 99% False positives: A qualitative study of SOC analysts' perspectives on security alarms. In *Proceedings of 31st USENIX Security Symposium (USENIX Security '22)*. USENIX Association, Boston, MA, 2783–2800. Retrieved from https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi

[8] Adham Albakri, Eerke Boiten, and Rogério De Lemos. 2019. Sharing cyber threat intelligence under the general data protection regulation. In *Privacy Technologies and Policy*. Maurizio Naldi, Giuseppe F. Italiano, Kai Rannenberg, Manel Medina, and Athena Bourka (Eds.), Springer International Publishing, Cham, 28–41.

[9] Adel Alshamrani, Sowmya Myneni, Ankur Chowdhary, and Dijiang Huang. 2019. A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Communications Surveys & Tutorials* 21, 2 (2019), 1851–1877. DOI: https://doi.org/10.1109/COMST.2019.2891891

[10] Roberto Andrade, Jenny Torres, and Susana Cadena. 2019. Cognitive security for incident management process. In *Proceedings of Information Technology and Systems (ICITS '19)*. Springer, 612–621.

[11] Frederik Armknecht, Youzhe Heng, and Rainer Schnell. 2023. Strengthening privacy-preserving record linkage using diffusion. *Proceedings on Privacy Enhancing Technologies* 2 (2023), 298–311.

[12] Tao Ban, Takeshi Takahashi, Samuel Ndichu, and Daisuke Inoue. 2023. Breaking alert fatigue: AI-assisted SIEM framework for effective incident response. *Applied Sciences* 13, 11 (2023), 6610.

[13] Vaclav Bartos, Martin Zadnik, Sheikh Mahbub Habib, and Emmanouil Vasilomanolakis. 2019. Network entity characterization and attack prediction. *Future Generation Computer Systems* 97 (2019), 674–686.

[14] Franziska Becker. 2021. *D3.9 Demonstrator of Visual Support for Designing Detection Models (Final Version)*. Technical Report. SAPPAN Consortium. Retrieved from https://sappan-project.eu/wp-content/uploads/2022/11/D3.9-Demonstrator-of-Visual-Support-for-Designing-Detection-Models-Final-version-M30.pdf

[15] Franziska Becker, Arthur Drichel, Christoph Müller, and Thomas Ertl. 2020. Interpretable visualizations of deep neural networks for domain generation algorithm detection. In *Proceedings of 2020 IEEE Symposium on Visualization for Cyber Security (VizSec)*, 25–29. DOI: https://doi.org/10.1109/VizSec51108.2020.00010

[16] Fabian Boemer, Rosario Cammarota, Daniel Demmler, Thomas Schneider, and Hossein Yalame. 2020. MP2ML: A mixed-protocol machine learning framework for private inference. In *Proceedings of the 15th International Conference on Availability, Reliability and Security (ARES '20)*. ACM, New York, NY, Article 14, 10 pages. DOI: https://doi.org/10.1145/3407023.3407045

[17] K. A. Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2016. Practical secure aggregation for federated learning on user-held data. In *Proceedings of NIPS Workshop on Private Multi-Party Machine Learning*, 5 pages. Retrieved from https://pmpml.github.io/PMPML16/papers/PMPML16_paper_8.pdf

[18] Xander Bouwman, Harm Griffioen, Jelle Egbers, Christian Doerr, Bram Klievink, and Michel van Eeten. 2020. A different cup of TI? The added value of commercial threat intelligence. In *Proceedings of 29th USENIX Security Symposium (USENIX Security '20)*. USENIX Association, 433–450. Retrieved from https://www.usenix.org/conference/usenixsecurity20/presentation/bouwman

[19] Xander Bouwman, Victor Le Pochat, Pawel Foremski, Tom Van Goethem, Carlos H. Ganan, Giovane C. M. Moura, Samaneh Tajalizadehkhoob, Wouter Joosen, and Michel van Eeten. 2022. Helping hands: Measuring the impact of a large threat intelligence sharing community. In *Proceedings of 31st USENIX Security Symposium (USENIX Security '22)*. USENIX Association, Boston, MA, 1149–1165. Retrieved from https://www.usenix.org/conference/usenixsecurity22/presentation/bouwman

[20] Juan Francisco Carías, Marcos R. S. Borges, Leire Labaka, Saioa Arrizabalaga, and Josune Hernantes. 2020. Systematic approach to cyber resilience operationalization in SMEs. *IEEE Access* 8 (2020), 174200–174221. DOI: https://doi.org/10.1109/ACCESS.2020.3026063

[21] Milan Cermak. 2020. *Stream-Based IP Flow Analysis*. Dissertation. Masaryk University. Retrieved from https://is.muni.cz/th/tgxb6/

[22] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM Computing Surveys* 41, 3, Article 15 (Jul. 2009), 58 pages. DOI: https://doi.org/10.1145/1541880.1541882

[23] Peter Christen, Thilina Ranbaduge, Dinusha Vatsalan, and Rainer Schnell. 2019. Precise and fast cryptanalysis for Bloom filter based privacy-preserving record linkage. *IEEE Transactions on Knowledge and Data Engineering* 31, 11 (2019), 2164–2177. DOI: https://doi.org/10.1109/TKDE.2018.2874004

[24] Peter Christen, Anushka Vidanage, Thilina Ranbaduge, and Rainer Schnell. 2018. Pattern-mining based cryptanalysis of Bloom filters for privacy-preserving record linkage. In *Advances in Knowledge Discovery and Data Mining*. Dinh Phung, Vincent S. Tseng, Geoffrey I. Webb, Bao Ho, Mohadeseh Ganji, and Lida Rashidi (Eds.), Springer International Publishing, Cham, 530–542.

[25] P. Cichonski, T. Millar, T. Grance, and K. Scarfone. 2012. *Computer Security Incident Handling Guide*, Vol. 800. NIST Special Publication, Gaithersburg, MD, 1–147 pages.

[26] Jason Creasey. 2013. *Cyber Security Incident Response Guide*. Crest.

[27] Cybersecurity and Infrastructure Security Agency. 2021. Cybersecurity Incident & Vulnerability Response Playbooks. Retrieved November 12, 2023 from https://www.cisa.gov/sites/default/files/publications/Federal_Government_Cybersecurity_Incident_and_Vulnerability_Response_Playbooks_508C.pdf

[28] Morten Dahl, Jason Mancuso, Yann Dupis, Ben Decoste, Morgan Giraud, Ian Livingstone, Justin Patriquin, and Gavin Uhma. 2018. Private machine learning in TensorFlow using secure computation. arXiv:1810.08130. Retrieved from http://arxiv.org/abs/1810.08130

[29] Anders Dalskov, Daniel Escudero, and Marcel Keller. 2020. Secure evaluation of quantized neural networks. *Proceedings on Privacy Enhancing Technologies* 4 (2020), 355–375.

[30] Fred J. Damerau. 1964. A technique for computer detection and correction of spelling errors. *Communications of ACM* 7, 3 (Mar. 1964), 171–176. DOI: https://doi.org/10.1145/363958.363994

[31] Arthur Drichel, Vincent Drury, Justus von Brandt, and Ulrike Meyer. 2021. Finding phish in a haystack: A pipeline for phishing classification on certificate transparency logs. In *Proceedings of the 16th International Conference on Availability, Reliability and Security (ARES '21)*. ACM, New York, NY, Article 59, 12 pages. DOI: https://doi.org/10.1145/3465481.3470111

[32] Arthur Drichel, Mehdi Akbari Gurabi, Tim Amelung, and Ulrike Meyer. 2021. Towards privacy-preserving classification-as-a-service for DGA detection. In *Proceedings of 2021 18th International Conference on Privacy, Security and Trust (PST)*, 1–10. DOI: https://doi.org/10.1109/PST52912.2021.9647755

[33] Arthur Drichel, Benedikt Holmes, Justus von Brandt, and Ulrike Meyer. 2021. The more, the better: A study on collaborative machine learning for DGA detection. In *Proceedings of the 3rd Workshop on Cyber-Security Arms Race (CYSARM '21)*. ACM, New York, NY, 1–12. DOI: https://doi.org/10.1145/3474374.3486915

[34] Arthur Drichel, Ulrike Meyer, Samuel Schüppen, and Dominik Teubert. 2020. Analyzing the real-world applicability of DGA classifiers *(ARES '20)*. ACM, New York, NY, Article 15, 11 pages. DOI: https://doi.org/10.1145/3407023.3407030

[35] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*. Shai Halevi and Tal Rabin (Eds.), Springer, Berlin, 265–284.

[36] Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407. DOI: https://doi.org/10.1561/0400000042

[37] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14)*. ACM, New York, NY, 1054–1067. DOI: https://doi.org/10.1145/2660267.2660348

[38] European Union Agency for Cybersecurity (ENISA). 2023. *ENISA Threat Landscape 2023*. DOI: https://doi.org/10.2824/782573

[39] Marina Evangelou and Niall M. Adams. 2020. An anomaly detection framework for cyber-security data. *Computers & Security* 97 (2020), 101941. DOI: https://doi.org/10.1016/j.cose.2020.101941

[40] Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Helen Möllering, Thien Duc Nguyen, Phillip Rieger, Ahmad-Reza Sadeghi, Thomas Schneider, Hossein Yalame, et al. 2021. SAFELearn: Secure aggregation for private FEderated learning. In *Proceedings of 2021 IEEE Security and Privacy Workshops (SPW)*, 56–62. DOI: https://doi.org/10.1109/SPW53761.2021.00017

[41] Gilberto Fernandes, Joel J. P. C. Rodrigues, Luiz Fernando Carvalho, Jalal F. Al-Muhtadi, and Mario Lemes Proença. 2018. A comprehensive survey on network anomaly detection. *Telecommunication Systems* 70, 3 (2018), 447–489. DOI: https://doi.org/10.1007/s11235-018-0475-8

[42] FIRST Standards Definitions and Usage Guidance. 2022. Information Exchange Policy (IEP). Retrieved December 6, 2023 from https://www.first.org/iep/

[43] FIRST Standards Definitions and Usage Guidance. 2022. Traffic Light Protocol (TLP)—Version 2.0. Retrieved December 6, 2023 from https://www.first.org/tlp/

[44] Inc Gartner. 2020. Security threat intelligence services reviews. Gartner. Retrieved November 12, 2023 from https://www.gartner.com/reviews/market/security-threat-intelligence-services

[45] Thomas Geras and Thomas Schreck. 2023. Sharing communities: The good, the bad, and the ugly. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23)*. ACM, New York, NY, 2755–2769. DOI: https://doi.org/10.1145/3576915.3623144

[46] Daniel Gibert, Carles Mateu, and Jordi Planes. 2020. The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. *Journal of Network and Computer Applications* 153 (2020), 102526. DOI: https://doi.org/10.1016/j.jnca.2019.102526

[47] Nitika Gupta, Issa Traore, and Paulo Magella Faria de Quinan. 2019. Automated event prioritization for security operation center using deep learning. In *Proceedings of 2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 5864–5872.

[48] Nicolas Heine. 2021. *Application Fingerprinting Based on System Events using Process Mining*. Master's thesis. RWTH Aachen University.

[49] Benedikt Holmes, Arthur Drichel, and Ulrike Meyer. 2021. Sharing FANCI features: A privacy analysis of feature extraction for DGA detection. In *Proceedings of 6th International Conference on Cyber-Technologies and Cyber-Systems (CYBER '21)*. IARIA XPS Press, 58–64.

[50] Martin Husák and Jaroslav Kašpar. 2018. Towards predicting cyber attacks using information exchange and data mining. In *Proceedings of 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 536–541.

[51] Martin Husák and Milan Čermák. 2022. SoK: Applications and challenges of using recommender systems in cybersecurity incident handling and response. In *Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES '22)*. ACM, New York, NY, Article 25, 10 pages. DOI: https://doi.org/10.1145/3538969.3538981

[52] Shouling Ji, Prateek Mittal, and Raheem Beyah. 2017. Graph data anonymization, de-anonymization attacks, and de-anonymizability quantification: A survey. *IEEE Communications Surveys & Tutorials* 19, 2 (2017), 1305–1326. DOI: https://doi.org/10.1109/COMST.2016.2633620

[53] Tomas Jirsik and Petr Velan. 2021. Host behavior in computer network: One-year study. *IEEE Transactions on Network and Service Management* 18, 1 (2021), 822–838. DOI: https://doi.org/10.1109/TNSM.2020.3036528

[54] Henrik Karlzen and Teodor Sommestad. 2023. Automatic incident response solutions: A review of proposed solutions' input and output. In *Proceedings of the 18th International Conference on Availability, Reliability and Security (ARES '23)*. ACM, New York, NY, Article 37, 9 pages. DOI: https://doi.org/10.1145/3600160.3605066

[55] Kathryn Knerler, Ingrid Parker, and Carson Zimmerman. 2022. *11 Strategies of a World-Class Cybersecurity Operations Center*. The MITRE Corporation.

[56] Dmitriy Komashinskiy and Andrew Patel. 2022. For security analysts, a picture may be worth more than a thousand words. Retrieved April 20, 2023 from https://sappan-project.eu/?p=1699

[57] Ifigeneia Lella, Eleni Tsekmezoglou, Marianthi Theocharidou, Erika Magonara, Apostolos Malatras, Rossen Svetozarov Naydenov, and Cosmin Ciobanu. 2023. *ENISA Threat Landscape 2023*. Technical Report. European Union Agency for Cybersecurity (ENISA). Retrieved from https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023

[58] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007. t-Closeness: Privacy beyond k-anonymity and l-diversity. In *Proceedings of 2007 IEEE 23rd International Conference on Data Engineering*, 106–115. DOI: https://doi.org/10.1109/ICDE.2007.367856

[59] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. 2023. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering* 35, 4 (2023), 3347–3366. DOI: https://doi.org/10.1109/TKDE.2021.3124599

[60] Vector Guo Li, Matthew Dunn, Paul Pearce, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. 2019. Reading the tea leaves: A comparative analysis of threat intelligence. In *Proceedings of 28th USENIX Security Symposium (USENIX Security '19)*. USENIX Association, Santa Clara, CA, 851–867. Retrieved from https://www.usenix.org/conference/usenixsecurity19/presentation/li

[61] Kun Liu and Evimaria Terzi. 2008. Towards identity anonymization on graphs. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD '08)*. ACM, New York, NY, 93–106. DOI: https://doi.org/10.1145/1376616.1376629

[62] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. 2007. L-Diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data* 1, 1, Article 3 (Mar. 2007), 52 pages. DOI: https://doi.org/10.1145/1217299.1217302

[63] Mandiant. 2013. OpenIOC 1.1. Retrieved December 6, 2023 from https://github.com/fireeye/OpenIOC_1.1

[64] Vasileios Mavroeidis, Pavel Eis, Martin Zadnik, Marco Caselli, and Bret Jordan. 2021. On the integration of course of action playbooks into shareable cyber threat intelligence. In *Proceedings of 2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2104–2108.

[65] Microsoft. 2023. *Microsoft Digital Defense Report 2023*. Retrieved from https://microsoft.com/mddr

[66] MISP. 2022. MISP Repository. Retrieved January 14, 2023 from https://github.com/MISP

[67] MITRE. 2012. CEE Language. Retrieved December 6, 2023 from https://cee.mitre.org/language/

[68] Pantaleone Nespoli, Dimitrios Papamartzivanos, Félix Gómez Mármol, and Georgios kambourakis. 2017. Optimal countermeasures selection against cyber attacks: A comprehensive survey on reaction frameworks. *IEEE Communications Surveys & Tutorials* 20, 2 (2017), 1361–1396.

[69] Lasse Nitz, Mehdi Akbari Gurabi, Avikarsha Mandal, and Benjamin Heitmann. 2021. Towards privacy-preserving sharing of cyber threat intelligence for effective response and recovery. *ERCIM NEWS* 126 (2021), 33–34.

[70] Lasse Nitz and Avikarsha Mandal. 2023. DGA detection using similarity-preserving Bloom encodings. In *Proceedings of the 2023 European Interdisciplinary Cybersecurity Conference (EICC '23)*. ACM, New York, NY, 116–120. DOI: https://doi.org/10.1145/3590777.3590795

[71] Lasse Nitz and Avikarsha Mandal. 2024. Bloom encodings in DGA detection: Improving machine learning privacy by building on privacy-preserving record linkage. *Journal of Universal Computer Science* 30, 9 (2024), 1224–1243. DOI: https://doi.org/10.3897/jucs.134762
[72] Lasse Nitz, Martin Zadnik, Mehdi Akbari Gurabi, Mischa Obrecht, and Avikarsha Mandal. 2022. From collaboration to automation: A proof of concept for improved incident response. *ERCIM NEWS* 129 (2022), 31–32.
[73] Don Norman. 2013. *The Design of Everyday Things: Revised and Expanded Edition*. Basic Books.
[74] Stephen Northcutt. 2003. *Computer Security Incident Handling—Step by Step Guide*. SANS Institute.
[75] Megan Nyre-Yu, Kelly A. Sprehn, and Barrett S. Caldwell. 2019. Informing hybrid system design in cyber security incident response. In *Proceedings of 1st International Conference on HCI for Cybersecurity, Privacy and Trust (HCI-CPT '19) Held as Part of the 21st HCI International Conference (HCII '19)*. Springer, 325–338.
[76] OASIS. 2016. CybOX^{TM} Version 2.1.1. Retrieved December 19, 2024 from https://docs.oasis-open.org/cti/cybox/v2.1.1/cybox-v2.1.1-part01-overview.html
[77] OASIS. 2019. Open Command and Control (OpenC2) Language Specification Version 1.0. Retrieved December 6, 2023 from https://docs.oasis-open.org/openc2/oc2ls/v1.0/oc2ls-v1.0.html
[78] OASIS. 2019. STIX Version 2.1. Retrieved December 6, 2023 from https://docs.oasis-open.org/cti/stix/v2.1/stix-v2.1.html
[79] OASIS. 2022. CACAO Security Playbooks Version 1.1. Retrieved January 14, 2023 from https://docs.oasis-open.org/cacao/security-playbooks/v1.1/security-playbooks-v1.1.html
[80] Ruoming Pang, Mark Allman, Vern Paxson, and Jason Lee. 2006. The devil and packet trace anonymization. *ACM SIGCOMM Computer Communication Review* 36, 1 (Jan. 2006), 29–38. DOI: https://doi.org/10.1145/1111322.1111330
[81] Daniel Plohmann, Khaled Yakdan, Michael Klatt, Johannes Bader, and Elmar Gerhards-Padilla. 2016. A comprehensive measurement study of domain generating malware. In *Proceedings of the USENIX Security Symposium*. Thorsten Holz and Stefan Savage (Eds.), USENIX Association, 263–278.
[82] Davy Preuveneers and Wouter Joosen. 2021. Sharing machine learning models as indicators of compromise for cyber threat intelligence. *Journal of Cybersecurity and Privacy* 1, 1 (2021), 140–163. DOI: https://doi.org/10.3390/jcp1010008
[83] Théo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. 2018. A generic framework for privacy preserving deep learning. arXiv:1811.04017. Retrieved from http://arxiv.org/abs/1811.04017
[84] D. Schlette, M. Caselli, and G. Pernul. 2021. A comparative study on cyber threat intelligence: The security incident response perspective. *IEEE Communications Surveys & Tutorials* 23, 4 (2021), 2525–2556. DOI: https://doi.org/10.1109/COMST.2021.3117338
[85] Daniel Schlette, Philip Empl, Marco Caselli, Thomas Schreck, and Günther Pernul. 2023. Do you play it by the books? A study on incident response playbooks and influencing factors. In *Proceedings of 2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 60–60.
[86] Sandra Schmitz-Berndt. 2023. Defining the reporting threshold for a cybersecurity incident under the NIS directive and the NIS 2 directive. *Journal of Cybersecurity* 9, 1 (2023), 11 pages. DOI: https://doi.org/10.1093/cybsec/tyad009
[87] B. Schneier. 2014. The future of incident response. *IEEE Security & Privacy* 12, 5 (Sep./Oct. 2014), 96.
[88] Rainer Schnell, Tobias Bachteler, and Jörg Reiher. 2009. Privacy-preserving record linkage using Bloom filters. *BMC Medical Informatics and Decision Making* 9, 1 (2009), 41.
[89] Rainer Schnell and Christian Borgs. 2016. Randomized response and balanced Bloom filters for privacy preserving record linkage. In *Proceedings of 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 218–224. DOI: https://doi.org/10.1109/ICDMW.2016.0038
[90] Samuel Schüppen, Dominik Teubert, Patrick Herrmann, and Ulrike Meyer. 2018. FANCI: Feature-based automated NXDomain classification and intelligence. In *Proceedings of 27th USENIX Security Symposium (USENIX Security '18)*. USENIX Association, Baltimore, MD, 1165–1181. Retrieved from https://www.usenix.org/conference/usenixsecurity18/presentation/schuppen
[91] Sebastian Schäfer and Ulrike Meyer. 2022. A pipeline for DNS-based software fingerprinting. arXiv:2208.07042. Retrieved from https://arxiv.org/abs/2208.07042
[92] Avi Shaked, Yulia Cherdantseva, and Pete Burnap. 2022. Model-based incident response playbooks. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, 1–7.
[93] Muhammad Sheeraz, Muhammad Arsalan Paracha, Mansoor Ul Haque, Muhammad Hanif Durad, Syed Muhammad Mohsin, Shahab S. Band, and Amir Mosavi. 2023. Effective security monitoring using efficient SIEM architecture. *Human-Centric Computing and Information Sciences* 13 (2023), 1–18.
[94] Jonathan M. Spring and Phyllis Illari. 2021. Review of human decision-making during computer security incident analysis. *Digital Threats: Research and Practice* 2, 2 (2021), 1–47.
[95] Latanya Sweeney. 2002. K-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 05 (2002), 557–570.
[96] Tines. 2022. *Voice of the SOC Analyst*. Retrieved April 4, 2024 from https://www.tines.com/reports/voice-of-the-soc-analyst
[97] Daniel Tovarnak and Martin Lastovicka. 2022. Malware Pipeline. Retrieved September 1, 2023 from https://github.com/CSIRT-MU/csirtmu-sappan-malware-evaluator

[98] Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. 2020. LDP-Fed: Federated learning with local differential privacy. In *Proceedings of the 3rd ACM International Workshop on Edge Systems, Analytics and Networking (EdgeSys '20)*. ACM, New York, NY, 61–66. DOI: https://doi.org/10.1145/3378679.3394533

[99] MuhammadFahad Umer, Muhammad Sher, and Yaxin Bi. 2017. Flow-based intrusion detection: Techniques and challenges. *Computers & Security* 70 (Sep. 2017), 238–254. DOI: https://doi.org/10.1016/j.cose.2017.05.009

[100] Anushka Vidanage, Thilina Ranbaduge, Peter Christen, and Rainer Schnell. 2019. Efficient pattern mining based cryptanalysis for privacy-preserving record linkage. In *Proceedings of 2019 IEEE 35th International Conference on Data Engineering (ICDE)*, 1698–1701. DOI: https://doi.org/10.1109/ICDE.2019.00176

[101] Manfred Vielberth, Fabian Böhm, Ines Fichtinger, and Günther Pernul. 2020. Security operations center: A systematic study and open challenges. *IEEE Access* 8 (2020), 227756–227779.

[102] Thomas D. Wagner, Khaled Mahbub, Esther Palomar, and Ali E. Abdallah. 2019. Cyber threat intelligence sharing: Survey and research directions. *Computers & Security* 87 (2019), 101589.

[103] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. 2020. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security* 15 (2020), 3454–3469. DOI: https://doi.org/10.1109/TIFS.2020.2988575

[104] Jun Xu, Jinliang Fan, M. H. Ammar, and S. B. Moon. 2002. Prefix-preserving IP address anonymization: Measurement-based security evaluation and a new cryptography-based scheme. In *Proceedings of 10th IEEE International Conference on Network Protocols*, 280–289. DOI: https://doi.org/10.1109/ICNP.2002.1181415

[105] ytisf. 2023. theZoo—A Live Malware Repository. Retrieved October 19, 2023 from https://thezoo.morirt.com/

[106] Mingchun Yun, Yuqing Lan, and Tao Han. 2017. Automate incident management by decision-making model. In *Proceedings of 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*. IEEE, 217–222.

[107] Martin Zadnik. 2021. Sharing response handling information. Retrieved December 19, 2024 from https://sappan-project.eu/wp-content/uploads/2022/11/D5.8-Sharing-Response-Handling-Information-M30.pdf

[108] Chuan Zhao, Shengnan Zhao, Minghao Zhao, Zhenxiang Chen, Chong-Zhi Gao, Hongwei Li, and Yu an Tan. 2019. Secure multi-party computation: Theory, practice and applications. *Information Sciences* 476 (2019), 357–372. DOI: https://doi.org/10.1016/j.ins.2018.10.024

[109] Bin Zhou, Jian Pei, and WoShun Luk. 2008. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *SIGKDD Explorations Newsletter* 10, 2 (Dec. 2008), 12–22. DOI: https://doi.org/10.1145/1540276.1540279

[110] Athanasios Zigomitros, Fran Casino, Agusti Solanas, and Constantinos Patsakis. 2020. A survey on privacy properties for data publishing of relational data. *IEEE Access* 8 (2020), 51071–51099. DOI: https://doi.org/10.1109/ACCESS.2020.2980235

Received 30 December 2023; revised 29 June 2024; accepted 12 November 2024