Universität Regensburg
Fakultät für Wirtschaftswissenschaften
Lehrstuhl für Wirtschaftsinformatik I - Informationssysteme

Cyber Threat Intelligence Exchange

Dissertation

Zur Erlangung des akademischen Grades “Doktor der Wirtschaftswissenschaft (Dr. rer. pol.)” an der Universität Regensburg gemäß der Promotionsordnung vom 23.07.2014, eingereicht an der Fakultät für Wirtschaftswissenschaften

vorgelegt von
Florian Menges, M.Sc.

Berichterstatter:
Prof. Dr. Günther Pernul
Prof. Dr.-Ing. Felix Freiling

To my parents, Günther and Maria,
and my fiancée Nadine

Abstract

The processing and exchange of Cyber Threat Intelligence (CTI) has become an increasingly important topic in recent years. This trend can be attributed to various factors. On the one hand, the exchange of information offers great potential to strengthen the knowledge base of companies and thus improve their protection against cyber threats. On the other hand, legislators in various countries have recognized this potential and translated it into legal reporting requirements. However, CTI is still a very young research area with only a small body of literature. Hence, there are hardly any guidelines, uniform standards, or specifications that define or support such an exchange. This dissertation addresses the problem by reviewing the methodological foundations for the exchange of threat intelligence in three focal areas. First, the underlying data formats and data structures are analyzed, and the basic methods and models are developed. In the further course of the work, possibilities for integrating humans into the analysis process of security incidents and into the generation of CTI are investigated. The final part of the work examines possible obstacles in the exchange of CTI. Both the legal environment and mechanisms to create incentives for an exchange are studied. This work thus creates a solid basis and a structured framework for the cooperative use of CTI.

Acknowledgement

In the past few years, a significant number of people have accompanied and supported me on my journey through the scientific world. Without their support, this work would not have been possible. Therefore, I would like to start this work by thanking all of you. First of all, I would like to thank my supervisor Prof. Dr. Günther Pernul in particular. He always lent his ear to my questions and problems, and was always there to give valuable feedback during my time at the chair and even afterwards. Moreover, he granted me the greatest possible freedom and thus provided me with the best possible environment to carry out my research. However, the most important factor for the success of my work was his precise sense of when a little pressure is needed to complete a paper. I would also like to thank my second supervisor, Prof. Dr.-Ing. Felix Freiling, for his valuable and constructive feedback during our mutual project meetings as well as in personal conversations. I would like to thank both generations of the IFS team, the Nexians, and the DINGfest project team for the warm and sincere collaboration—you were the best colleagues and friends that anyone could wish for. I would like to express my gratitude to Ludwig, Hannes, and Stefan who taught me a lot about writing papers. I am also grateful to Mr. Weber who taught me about the importance of careful and diligent correction of papers and to Hasi for his lessons on independence. I would like take this opportunity to thank the outstanding colleagues of the current IFS generation, I was privileged to work with. Robber Baron Böhm certainly deserves the biggest thanks for his unfailingly reliable support in the research project. But I also want to extend my thanks to the rest of the team—Bene and Dani for the countless Blockchain-, CTI, and market discussions; Ms. Weber for taking over Mr. Weber's position with unsurpassed skill; Lucke's lessons in track alignment; Pferdl for playing the devil's advocate in our discussions; Christine for countless data format discussions, and Huaba and Petra for their technical, corrective, and emotional support over the years. I would also like to thank the entire DINGfest team for the smooth and fruitful cooperation in our joint research. My sincerest thanks go to the Federal Ministry of Education and Research for providing financial support to my research within the DINGfest project. A very central role was also played by Michi, Sichermut and Matthias who have been with me during intense periods of research and private concerns. Matthias's reality checks were particularly noteworthy—I owe you a drink. But above all, I would like to thank my parents, Günther and Maria, and my fiancée, Nadine, for your tireless support, your belief in me, as well as your patience during all these years

Contents

Abstract **i**

Acknowledgement **ii**

List of Tables **iv**

List of Figures **vi**

I Overview of the Thesis **vii**

1 Introduction 1

2 Research Questions 4

3 Research Methodology 7

4 Results 11

4.1 Overview of Research Papers 11

4.2 Structured Representation of CTI 12

4.3 Integration of human beings in CTI processing 17

4.4 Overcoming obstacles in CTI exchange 23

4.5 Complementary publications 28

5 Conclusion and Future Work 29

II Research Papers **32**

1 A comparative analysis of incident reporting formats 32

2 Unifying Cyber Threat Intelligence 48

3 Graph-based visual analytics for cyber threat intelligence 64

4 Human-as-a-security-sensor for harvesting threat intelligence 84

5 Towards GDPR-compliant data processing in modern SIEM systems . . 100

6 Incentives for Threat Intelligence Reporting and Exchange 135

References **151**

List of Tables

1 Overview of research papers within this dissertation 11
2 Overview of complementary research papers 29

List of Figures

1 CTI lifecycle according to Chantzios et al. [5] 2

2 Process model according to Peffers et al. [19] 9

3 Overview of research papers and corresponding research areas 12

4 Incident detection process based on Paper P1 13

5 Universal pattern for structured incident exchange based on Paper P1 . . 14

6 CTI meta model based on Paper P2 15

7 Unified CTI data model based on Paper P2 16

8 KAVAS persistence mechanism based on Paper P3 18

9 KAVAS user interface based on Paper P3 19

10 Incident model and taxonomy based on Paper P4 21

11 Human-as-a-security-sensor wizard based on Paper P4 22

12 GDPR-compliant SIEM architecture based on Paper P5 24

13 Decentralized ecosystem based on Paper P6 26

14 DEALER user interface based on Paper P6 27

Part I

Overview of the Thesis

1 Introduction

Information technology (IT) has advanced into nearly all areas of life in recent years and has become an indispensable part of our daily lives. It has also become a driver of growth and a key technology for companies and institutions, making a significant contribution to secure their competitiveness. With the widespread use and increasing importance of IT, it is also continuously becoming a target of cybercrime. The attacks are growing in their complexity and intensity, and sometimes cause immense damage [17]. A wide variety of scenarios can be observed, ranging from data theft to ransom and sabotage. The threat situation described is particularly problematic if the attacks affect critical infrastructures. These are systems and assets fundamental to the functioning of society and whose disruption or failure could have dramatic consequences for public order and human security [6].

In order to ensure protection against the dangers of cybercrime, a variety of methods and approaches have been developed and published in recent years. This includes a large number of possible measures to detect and prevent future attacks as well as defend and mitigate the impacts of damaging events that have already occurred. The arsenal of countermeasures ranges from access control procedures and firewalls to the detection of attacks and emergency recovery plans. These traditional approaches, however, often work with an insufficient and isolated knowledge base; they only provide reliable protection against threats which are already known [26]. As a result, there is often little protection against or situational awareness of current threats, and defensive mechanisms can often only be initiated reactively.

The problems outlined above have also been recognized by companies and nation states [23]. As a result, there is a recent trend toward collaboration in the defense against cyber threats. For example, the first collaborative projects have already emerged between companies to jointly increase their security level by exchanging information on security incidents and threats[1],[2]. Approaches involving the exchange of information on threats are also encouraged and supported by different governments. More specifically, various laws have been passed in this regard, such as the IT Security Act in Germany [7] and the NIS Directive 2016/1148 in the European Union [10]. These regulations define a number of reporting obligations in cases of suspected and actual damage, mostly in connection with critical infrastructures. Apart from its practical relevance, the beneficial effects of the exchange of information are widely recognized within scientific literature and have been discussed in various publications [28, 21, 4].

At the center of information exchange for cooperative security are the data structures which allow identified threats to be stored and transported. In the literature, this is commonly referred to as Cyber Threat Intelligence (CTI). Although there is no uniform and sharp definition of CTI to date, a work by the Bank of England from 2016 provides a good starting point. Following this, CTI can be considered as "information about threats

[1] https://www.allianz-fuer-cybersicherheit.de

[2] https://www.blocklist.de

and threat actors that provides a sufficient understanding of the containment of harmful events" [2]. Chantzios et al. [5] go one step further and specify existing definitions by describing CTI as something that goes beyond the mere description of data structures that represent threats. According to this work, CTI allows the description of a process representing the entire lifecycle of information processing that leads to an understanding of threat situations. This process includes the identification of data sources, the acquisition of data, the analysis of acquired data, their exchange, and a final review of continuous improvement of the process, and it is outlined in Figure 1. With the processing and exchange of CTI, several positive effects can be achieved. These include, for example, a higher situational awareness of those involved, a deeper knowledge of threats, and improved defense capabilities.

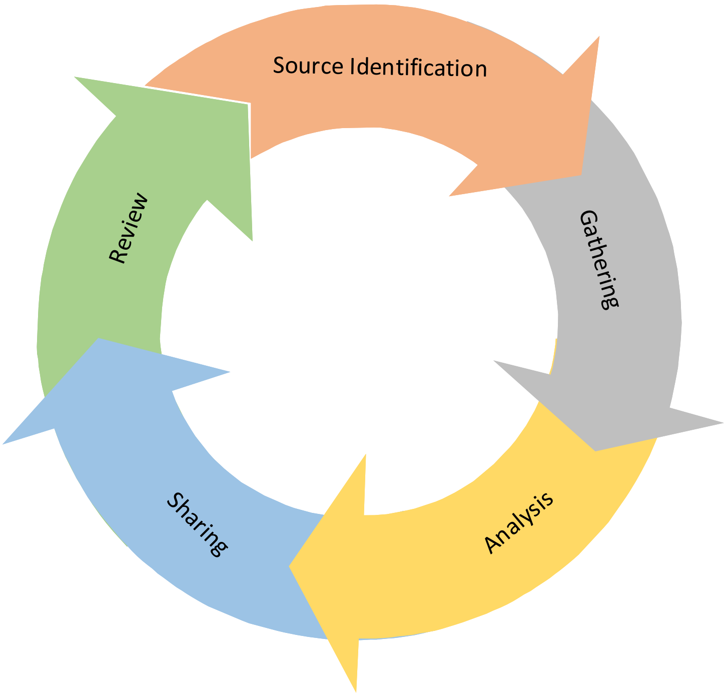


Figure 1: CTI lifecycle according to Chantzios et al. [5]

While CTI is a promising technology for increasing the level of protection, it nevertheless remains a very young field of research. As a result, its usage is accompanied by various challenges and problems of implementation. A major obstacle in the exchange of information is already apparent in the structuring of CTI. Existing approaches often differ greatly from each other, suffer from incompatibilities, or demand a commitment to proprietary software. In addition, existing data structures often only support low-level information, while information on a semantically higher level cannot be represented. The inclusion of human knowledge is also a major problem in this context. Although employees may possess valuable operational and strategic knowledge, the interaction of human beings with the CTI lifecycle has hardly been investigated so far. Moreover, exchanging information is always connected to balancing the costs against the benefits for companies, which can lead to information not being exchanged. Despite being well-known and highly relevant for companies owing to legal requirements, solution approaches have hardly been discussed in the literature so far. In this field of tension, the goal of this dissertation is to address the problems described above and clarify important fundamental questions in the field of threat intelligence processing and exchange. For this purpose, the research carried out first examines the structuring possibilities of CTI. As a result, important findings are provided that contribute to increasing compatibility in the exchange and to developing a better understanding of the information among the participants. Building on this, the possibilities of human interaction with this structured information are investigated. This creates the basis for introducing contextual information into automated analysis processes and thus considerably increasing the meaningfulness of the information. Based on these findings, the dissertation finally proposes approaches to create incentives for the exchange in the context of legally compliant reporting, thus laying the foundation for a sustainable exchange.

2 Research Questions

The exchange of threat intelligence is still a relatively young and barely researched discipline in science. As a result, various problems, obstacles, and uncertainties can be expected in its utilization. The central aspect of an exchange is the content of the information to be transmitted. Only if the transmission is carried out in a uniform, comprehensible, and meaningful form can the recipient make use of it. The scope and complexity of CTI raises another problem. In order to be able to generate a benefit from transmitted data, uniform access must be provided to evaluate the information contained therein. In addition, such an exchange is associated with high costs and various legal requirements. Legal reporting obligations must be complied with and legal requirements for data protection must be taken into account. These factors and problems lead to the central research question (RQ) for this dissertation:

> **RQ:** *How can Cyber Threat Intelligence be harnessed through a structured exchange while overcoming exchange barriers?*

This central research question is broken down and answered through six partial research questions across three research areas. In the first area, possibilities for the structured presentation of CTI are examined. Based on this, the second area examines the possibilities for involving individuals in the CTI operating process. Finally, in the third area, possibilities for counteracting potential obstacles that may impede an exchange are examined.

Structured representation of CTI

The first focus area of this work is the challenge of how CTI can be stored, represented, and exchanged in a structured way. This is an essential factor for developing a common understanding and further processing of the content. In practice, various exchange formats and data structures already exist, such as Structured Threat Information eXpression (STIX) [3], Incident Object Description Exchange Format (IODEF)[3], Vocabulary for Event Recording and Incident Sharing (VERIS)[4], and Malware Information Sharing Platform and Threat Sharing (MISP) [27]. Although these data formats have been developed largely independent of each other by different actors, all of them pursue the general goal of providing comprehensive information about threats and incidents. Due to their origin in different use cases and requirements, however, the underlying data structures often differ considerably. They are already used to a limited extent in the security operations of some companies and organizations. Because different data formats are used in different versions, the exchange of data across company boundaries often poses a big challenge. In the literature, studies have described some of these data formats in detail, including the works of [25], [1], and [12]. However, in-depth analyses and qualitative comparisons

[3] https://tools.ietf.org/html/rfc5070

[4] http://veriscommunity.net/

of these formats have not been carried out yet. This problem leads to the first partial research question RQ1.

> **RQ1:** *How can CTI exchange formats be described and compared structurally and qualitatively?*

A further problem in this context arises from the differences between the individual data formats. These are often the result of different requirements and different use cases which lead to significant differences in syntax and semantics. Owing to these significant differences in their characteristics, the data formats are often incompatible, making cross-format data exchange very difficult. This problem leads to the second partial research question RQ2.

> **RQ2:** *Which essential characteristics define CTI exchange formats and how can they be standardized?*

Integration of human beings in CTI processing

The second focal area of this dissertation is to provide individuals with access to the information exchanged. Existing data formats usually use extensive serialization mechanisms based on XML or JSON. These are generally not accessible for humans, especially if they contain large amounts of data. Besides, the automated evaluation of threat information quickly reaches its limits when the data to be analyzed is too extensive. In such cases, a subsequent analysis by security experts is often indispensable. This requirement results in the partial research question RQ3.

> **RQ3:** *How can experts be integrated into a CTI analysis and exchange process?*

In addition to dedicated analyses by security experts, the creation of a human-machine interface for CTI opens up further opportunities. It also results in a great potential in the area of data acquisition. Existing systems typically rely on the automatic collection and analysis of threat intelligence from log files. With the integration of human actors and their knowledge into the acquisition process, there is the potential to extend the scope of existing CTI significantly. These considerations result in the partial research question RQ4.

> **RQ4:** *How can employees with different knowledge levels be integrated into the CTI exchange process?*

Overcoming obstacles in CTI exchange

The third focal area of the work deals with possibilities to overcome potential obstacles in the exchange of threat intelligence. Such exchanges are subject to legislation in different countries which must be taken into account. This includes reporting obligations such as those stipulated by the German IT security (ITSiG) law on the one hand and data protection requirements, for example, as defined by the European General Data Protection Regulation (GDPR) on the other hand. These requirements and the resulting obstacles lead to the partial research question RQ5.

> **RQ5:** *How can reporting obligations be fulfilled using CTI while complying with applicable legislation?*

In addition to possible obstacles due to the legal requirements, companies may also face certain disadvantages and problems which arise from an exchange. Specifically, the analysis and exchange of information about security incidents results in high costs, such as through the provision of the necessary infrastructure and the corresponding specialist personnel. Furthermore, an exchange always involves the risk of data leakage, which could result in further costs. This may lead to a situation where companies are only interested in passive participation in the exchange, as the risks of an active exchange may be considered disproportionate. This problem leads to the partial research question RQ6.

> **RQ6:** *How can the exchange of CTI be incentivized?*

Overall, the answers to the partial research questions RQ1-RQ6 and thus, to the central research question RQ are intended to provide the scientific basis for an exchange of threat intelligence. This dissertation therefore provides the necessary tools for the analysis and evaluation of CTI data structures, the development of interfaces for humans to this technology, and the minimization of possible obstacles in its use.

3 Research Methodology

The previously defined research questions are answered in this dissertation with the use of established research methods from the field of *Wirtschaftsinformatik*. For the purpose, this section first classifies the research area of *Wirtschaftsinformatik* and important research methods. Building on this, the applied research methodology is presented and the specific application within this work is discussed.

The discipline of *Wirtschaftsinformatik* is often translated as "business and information systems engineering," but the actual equivalent in the Anglo-Saxon world is the field of "information systems research." Here, the two main research paradigms "behavioral science" and "design science research" are applied [18]. **Behavioral science** has its origins in the natural sciences. It mainly studies the explanation and prediction of phenomena in the areas of analysis, design, implementation, and management of information systems. The goal is to inform researchers and practitioners about interactions among people, technology, and organizations to find out the "truth" about IT systems. This is meant to ensure that IT systems can fulfill their purpose of increasing the efficiency of organizations [13]. The approach of **design science research**, on the other hand, has its roots in the field of engineering and fundamentally describes a problem-solving paradigm. In this context, innovations should be generated on the basis of ideas, practices, technical skills, or products to enable the efficient use of information systems. Specifically, existing theories are applied, tested, modified, and extended to create artifacts. These are then used to understand and solve problems [13]. The two research paradigms of information systems research are used differently in the scientific community. While behavioral science is the predominant research paradigm in the Anglo-Saxon community, the German research community is more design-oriented [18]. Consequently, this dissertation is also significantly influenced by the design science paradigm. More precisely, the work follows the basic guidelines for design science as put forward by Hevner et al. [13] as well as the methodological framework for design science proposed by Peffers et al. [19]. In the following, the applied approaches are described in more detail with reference to the contents of this work. According to Hevner et al., the fundamental principle of design science research is to develop and apply an artifact to solve a problem [13]. The goal is to create an understanding of and knowledge about a design problem and to solve it. Based on this idea, Hevner et al. propose the following seven guidelines for design science research [13].

Guideline 1. Design as an artifact

A central requirement for design science research is the creation of a useful and feasible artifact to solve an important organizational problem. According to Hevner et al., artifacts are defined as constructs, models, methods, or instantiations. They can manifest as abstractions, representations, models, methods, algorithms, or specific practices. Such artifacts are developed for each research question of this work's three focal areas. Each should contribute to the solution of the respective sub-problem and address the central research question in its entirety. In order to ensure the usefulness of the artifacts developed, each of the focal areas of the work sets its own development focus. For example, questions related to the representation of CTI are primarily answered by constructs, models, and methods, while the integration of the human component and the addressing of exchange obstacles are addressed by models and instantiations.

Guideline 2. Problem relevance

A further requirement in design science research is to develop technology-based solutions that are important and relevant for previously unsolved business problems. The dissertation deals mainly with research questions from the relatively young research field of threat intelligence sharing, which has become increasingly important in recent years [22]. The problems addressed may have already been identified as relevant in the literature, show practical relevance in the business environment, or can be justified by the current legislation. The relevance of the addressed problems and research questions is described in detail in each of the research papers.

Guideline 3. Design evaluation

The design evaluation guidelines provide for the evaluation of the developed artifact in terms of utility, quality, and effectiveness. The evaluation is to be performed using established procedures and methods. The research papers produced in the course of this work each contain dedicated evaluations of the results obtained. Different evaluation methods were used, ranging from *observational* to *analytical* and *descriptive*. In addition to the evaluations, the individual papers contain justifications for the selection of the method used.

Guideline 4. Research contributions

Design science research tackles the question of what new and interesting contributions could be made. A contribution can be the artifact itself, but it could also be an extension to the knowledge base or an evaluation methodology. In the context of this dissertation, several artifacts are provided as separate contributions — for example in the form of prototypical applications. In addition, several contributions to the knowledge base are made. This includes, for example, new findings in the area of CTI data formats and an evaluation methodology for CTI data formats.

Guideline 5. Research rigor

Another essential part of design science research is the application of strict rules to the construction and the evaluation of the developed artifact. It is important to apply the available knowledge proficiently to create and evaluate an artifact using suitable methods. The publications resulting from this work rely strictly on the available knowledge base from the literature for the development of concepts, methods, and prototypes. For the results obtained in this process, appropriate evaluation procedures are carefully selected and applied.

Guideline 6. Design as a search process

Design can be described as a process for efficient problem-solving, taking into account available resources, pursued objectives, and environmental conditions. Existing problems

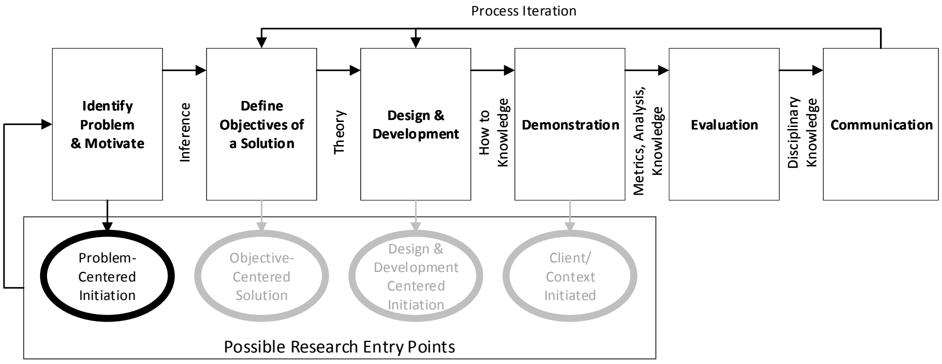

Process Iteration
Identify Problem & Motivate
Inference
Define Objectives of a Solution
Theory
Design & Development
How to Knowledge
Demonstration
Metrics, Analysis, Knowledge
Evaluation
Disciplinary Knowledge
Communication
Problem-Centered Initiation
Objective-Centered Solution
Design & Development Centered Initiation
Client/ Context Initiated
Possible Research Entry Points

Figure 2: Process model according to Peffers et al. [19]

are often simplified or subdivided into sub-problems by reducing them to sub-areas, which was also consistently pursued in this dissertation. The identified overall problem is defined as a research question and split into three focal areas. Within these areas, the individual research questions are covered by publications, which in turn are divided into sub-problems. The solution of the defined sub-problems should finally lead to the solution of the formulated overall problem.

Guideline 7. Communication of research

Research results from design science research are to be presented to both technology- and management-oriented audiences. Technical presentations should highlight the advantages of the artifact, the possibilities for expanding the scientific knowledge base, and the environmental conditions under which the artifact was created. Management presentations, on the other hand, should convey how an artifact can be used in organizational contexts. The papers produced in the context of this dissertation offer extensive information for a technical audience, while also addressing organizational applications where possible. All resulting papers have been submitted or published in scientific journals and conferences. The results have also been directly incorporated into the DINGfest research project[5]. Thus, the research results are made accessible to a broad technically and management-oriented audience.

The guidelines of Hevner et al. presented here describe the characteristics of well-performed research and form the methodological foundation of this work. The applied research process is based on these characteristics and follows the *design science research methodology* and process model proposed by Peffers et al. [19]. Figure 2 outlines the essential components of this process model which consists of six activities and four possible entry points.

The research process begins in the first step with the identification of the problem and the motivation. Here, the relevance of the presented problem as well as the necessity and benefits of the solution are clarified. In the second step, specific goals for a solution are defined. It either describes how a solution improves the state-of-the-art or how a new

[5] https://dingfest.ur.de

artifact can contribute to the solution. The third step describes the design and development of an artifact including the actual research contribution. After the artifact has been created, the fourth step shows that it can solve one or more instances of the defined problem. On this basis, an evaluation of the results can be performed in the fifth step. In doing so, it can be observed or measured how well the artifact actually supports the solution to the addressed problem. Finally, in the sixth step, the results of the research are communicated. For this purpose, the problem and its relevance, benefits, novelty, design stringency, and efficiency of the solution are communicated to researchers and other relevant target groups. Since the process model is not built in a continuously linear fashion, it allows the return to previous process steps and thus the multiple execution of process steps.

The model also provides possible entry points into the first four phases of the model. Due to the problem-oriented structure of the publications within this dissertation, the entry points are essentially located at process step 1. In the following chapter, the specific research results of the individual works are presented.

4 Results

4.1 Overview of Research Papers

The research questions posed in Section 2 were answered by a total of six research papers which were published together with this dissertation according to the guidelines and procedure model described in Section 3. To ensure that the results could be communicated to a suitable specialist audience, all the papers have been submitted to renowned specialist journals and conferences in the fields of IT security and information systems. An overview of the individual research papers produced in this context can be found in Table 1. It shows the six research papers listed in the order of the corresponding research questions outlined in Section 2. Each paper is directly assigned to the respective research questions RQ1-RQ6 by their numbering P1-P6. However, the order does not reflect the chronological publication dates of the articles. In addition to the numbering assignments, the table presents the full citation and the current submission status of the article. Moreover, the type of each article is specified, indicating whether the contribution was submitted for publication at a conference (C) or in a journal (J). At the time of writing the dissertation, Papers 1-4 have been accepted and published, while Papers P5 and P6 are currently still subject to a review process. The complete papers, more detailed information on the submissions, and the proportion of contribution by the authors to the respective articles are provided in Part II of this dissertation.

No.	Publication	Status	Type
P1	Menges, F. and Pernul, G. A comparative analysis of incident reportingformats. In: *Computers & Security 73*, 87–101 (2018).	published	J
P2	Menges, F., Sperl, C., and Pernul, G. Unifying cyber threat intelligence. In: *Trust, Privacy and Security in Digital Business, TrustBus*. LNCS, vol. 11711, pp. 161–175. Springer, Cham (2019)	published	C
P3	Böhm, F., Menges, F., and Pernul, G. Graph-based visual analytics for cyber threat intelligence. In: *Cybersecurity 1*, 16 (2018).	published	J
P4	Vielberth, M., Menges, F., and Pernul, G. Human-as-a-security-sensor for harvesting threat intelligence. In: *Cybersecurity 2*, 23 (2019).	published	J
P5	Menges, F., Latzo, T., Vielberth, M., Sobola, S., C. Pöhls, H., Taubmann, B., Köstler, J., Puchta, A., Freiling, F., Reiser, H. P., and Pernul, G. Towards GDPR-compliant data processing in modern SIEM Systems. *Computers & Security* (2020)	under review	J
P6	Menges, F., Putz, B., and Pernul, G. DEALER: Decentralized Incentives for Threat Intelligence Reporting and Exchange, *International Journal of Information Security* (2020)	under review	J

Table 1: Overview of research papers within this dissertation

As shown in Section 2, the dissertation consists of three consecutive topics. Figure 3 gives a graphical overview of the main topics and the categorization of the individual research papers within these topics.

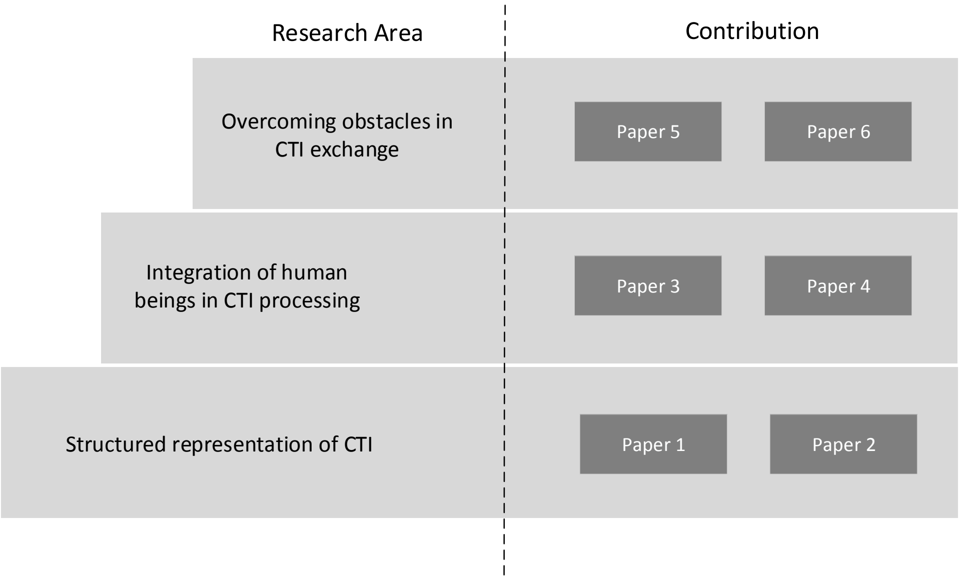

Research Area
Contribution
Overcoming obstacles in CTI exchange
Paper 5
Paper 6
Integration of human beings in CTI processing
Paper 3
Paper 4
Structured representation of CTI
Paper 1
Paper 2

Figure 3: Overview of research papers and corresponding research areas

The foundation for this dissertation is formed through the investigation of possibilities for the structured representation of CTI with papers P1 and P2. Building on this, the second focal area of the work is the creation of interfaces for humans to access the structured CTI. To achieve this, the integration possibilities of domain experts as well as of employees without IT security knowledge are examined in papers P3 and P4. On the basis of this preliminary work, the final phase of the research examines how obstacles in the exchange of threat intelligence information can be overcome. More specifically, legal implications for CTI exchange procedures and possibilities for creating reporting incentives are examined in papers P5 and P6. The results of the individual research areas are presented in detail below.

4.2 Structured Representation of CTI

At the center of any exchange of threat intelligence are the data structures used to store and transmit the information. Their specific properties, such as syntax, semantics, and unambiguousness of the contents or machine readability, are essential influencing factors for the usability of the data as well as its comprehensibility in the context of an exchange. The compatibility of different data structures with each other is also a necessary condition for any exchange process. For these reasons, it is essential that the data formats and data structures used in an exchange are known, well-investigated, uniform, and understandable for all participants. In order to provide such a methodical foundation, the first focus of this work is to analyze existing CTI data structures and investigate the potential for standardizing existing data formats.

P1: A comparative analysis of incident reporting formats

Paper P1 of this dissertation describes the structural foundations of CTI data structures and creates an evaluation basis for the components of threat intelligence. The result of this work serves to answer research question RQ1 (see Section 2) and shows how CTI exchange formats can be described and compared structurally as well as qualitatively.

An essential problem in the investigation of CTI data structures is that different data formats are used for different application purposes. These range from data formats which represent vulnerabilities and formats for data exchange between intrusion detection and prevention systems to formats for the representation of complex attacks and security incidents. In addition, the data structures used, even if the same application purpose is pursued, sometimes differ significantly in syntax, semantics, and scope.

In order to enable the classification and comparability of threat intelligence data formats, a multi-layered approach was followed in this paper. First, the most important CTI data formats were identified, described, and classified. This made it possible to establish a distinction to other IT security-related data formats. On this basis, in a second step, essential features of the identified data formats were determined, translated into models, and the criteria for qualitative comparisons were derived. In the final step, these tools were used to perform a structural and qualitative comparison of the exchange formats. The initial classification of existing CTI data structures was carried out in this work by identifying essential data types within an analysis process of security incidents. Individual data formats within the data analysis were identified and translated into a process. The process is outlined in Figure 4 and provides an overview of the development of CTI data structures along the data analysis process. The process shows how unstructured raw data is first acquired and converted into structured CTI data. The process comprises several stages, beginning with the transfer of the **raw data** into partially structured **actionable observables**, through the detection of **indications** of security incidents, to the transfer of the information into fully structured CTI **incident** data. Enrichments using **enumeration** objects, which can avoid ambiguities in the information exchanged, are also taken into account within the process.

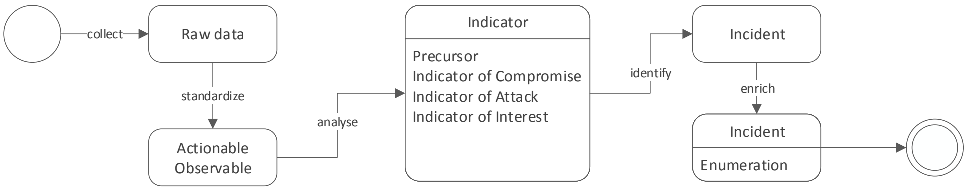


Figure 4: Incident detection process based on Paper P1

Based on these data structures, the next step of this work was to develop a pattern for the generic representation of incident data structures. The resulting Universal pattern for structured incident exchange (UPSIDE) shown in Figure 5 represents the essential CTI data points on an object level.

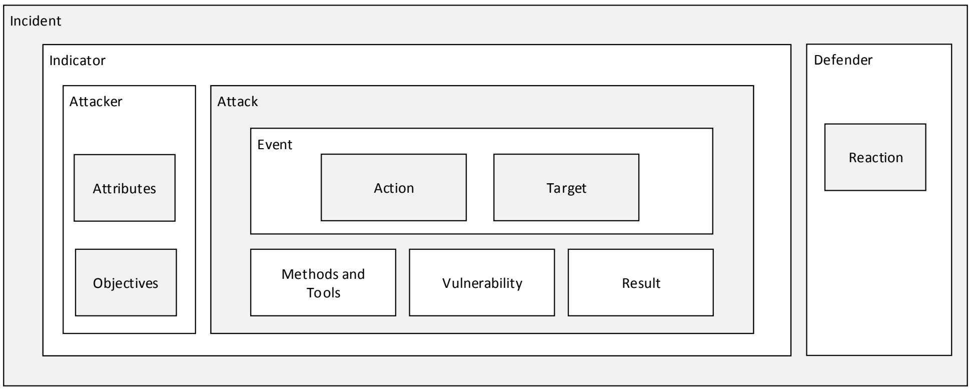


Figure 5: Universal pattern for structured incident exchange based on Paper P1

The starting point for this model was the existing scientific basis for CTI data formats from the literature and established data formats from practice. Existing data formats generally use nested XML or JSON data structures for data handling. In order to enable an accurate representation of the data formats, a nested representation was also chosen for the creation of the UPSIDE model. The model illustrates the relationship shown in the process model — namely, that an incident object is composed of different indicators. Indicators, in turn, transport the actual user data about the represented security incident. This contains detailed information about the attacker involved, the attack, and the methods used. In addition to indicators, information about the behavior and possible countermeasures of the defender can be transported. Based on these findings, various criteria for comparing CTI data formats were developed in the next step. On the one hand, these include structural criteria resulting from the models presented. On the other hand, qualitative criteria were developed which could be derived from literature, practice, and the characteristics of existing data formats. Finally, a comparison of the most important CTI data formats was carried out using the criteria determined here, thus providing a comprehensive overview of the state-of-the-art in CTI. With this analysis, the different strengths and weaknesses of the analyzed data formats could be revealed. It turned out that the STIX data format currently offers the most extensive representation possibilities, while soft criteria such as extensibility showed a mixed picture.

Contribution of P1:
In summary, this work provided the basis for the identification, description, and comparison of CTI data structures. It was possible to classify the corresponding data structures within the analysis process and to develop a pattern for a generic description of threat intelligence. In addition, criteria were developed for comparing CTI data formats. The application of these criteria has finally made it possible to compare the state-of-the-art data formats.

P2: Unifying Cyber Threat Intelligence

The comparability of CTI data structures has been established using Paper P1. Next, Paper P2 focuses on the exploration of potentials for the standardization of these data structures. While Paper P1 considers CTI data structures on the object and data format level, Paper P2 examines data formats on the attribute and data type level. It describes the structural properties of CTI exchange formats in detail and presents a methodology for their standardization to address the research question RQ2 (see Section 4.2).

A major problem in the exchange of CTI is that different organizations and companies have already integrated selected data formats into their processes and operational use. This includes data formats such as STIX, IODEF, and VERIS, which are used for the structured storage of CTI and have become widely used in recent years. To complicate the situation even more, these data formats are often used in different versions in practice. As a result of these differing data structures, an exchange across company boundaries is often difficult or even not possible at all.

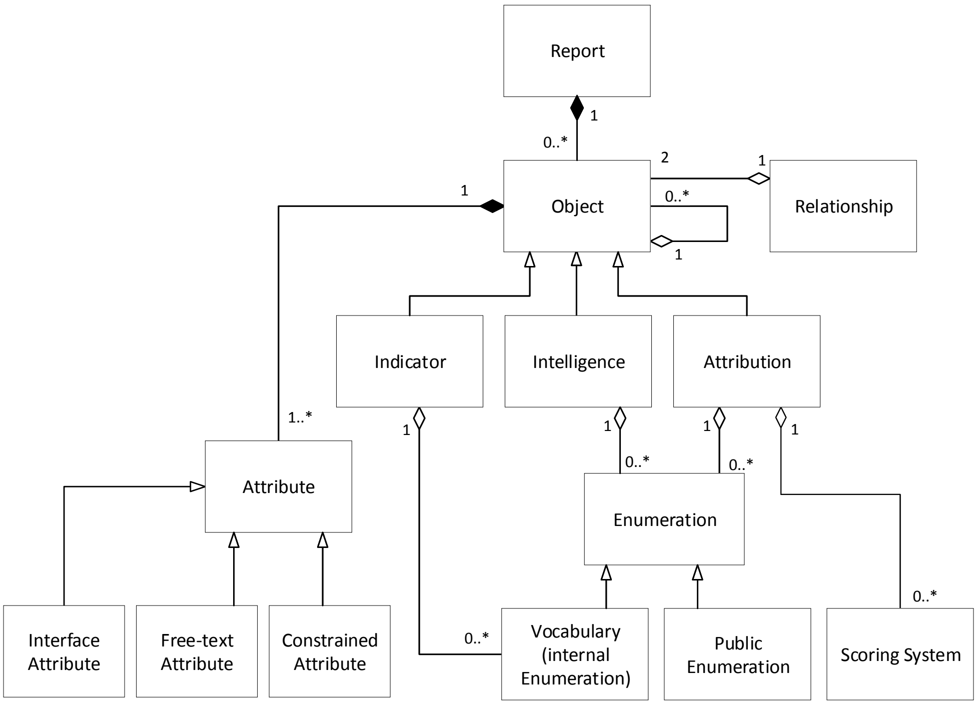


Figure 6: CTI meta model based on Paper P2

This paper addresses the question of how such incompatibilities can be overcome by creating a uniform data structure for the exchange. To make this possible, first an in-depth investigation of the components within CTI data structures based on the results of Paper P1 is carried out. In contrast to P1, where the focus was on the description of CTI entities at the object level, P2 considers CTI data formats at the structural level. In order to provide a complete and detailed picture of CTI data structures, the relationships and attributes included are examined in detail. To achieve this structural view, a CTI meta model was developed in this paper. It represents the characteristics of CTI data structures and serves as a starting point for a standardization of different data formats. The model is based on findings from the literature on the one hand and on properties of data formats from practice on the other. From a methodological perspective, it is based on the abstraction concepts for the creation of meta models as per Sprinkle et. al. [24] and is shown in Figure 6.

The model represents the possible structural relationships among objects, relations, and attributes within CTI exchange formats. The three main CTI entity classifications — **indicator**, **intelligence** and **attribution** have been translated into specific objects within the model. In addition, the essential attribute types — **attribute**, **enumeration** and **scoring system** — were identified and assigned to the individual objects within the model. Furthermore, the model provides information on possible **relationships** between the individual entities of the model. The structural properties of CTI data formats presented here are intended to ensure re-usability of the results on the one hand and to leave no room for interpretation when investigating CTI data formats on the other.

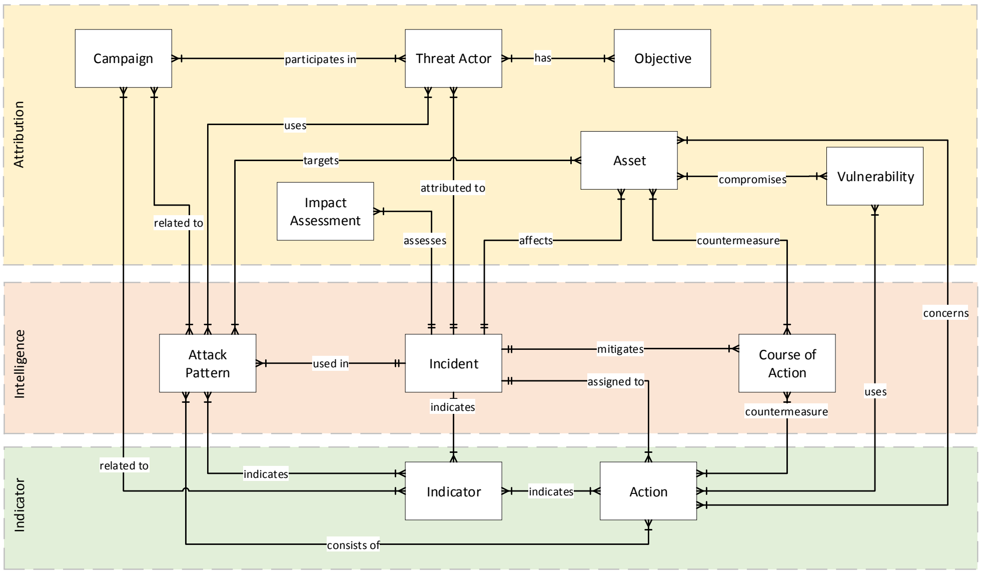


Figure 7: Unified CTI data model based on Paper P2

In addition to the meta model, a methodology for the standardization of CTI entity types was also developed in this work. For this purpose, a set of rules was defined which allows to convert CTI data formats into a uniform and reproducible notation. In this process, the essential CTI entity types were identified from the state-of-the-art CTI data formats and converted into a uniform notation with the help of the rule set.

In the last step of the work, the knowledge gained from the previously developed meta model and the unified CTI notation was used to develop a unified CTI data model. The model is designed as an entity relationship model and shown in Figure 7.

The model integrates all CTI entity types identified in the course of this work and labels each of them using the unified notation. In addition, the previously defined classifications of **attribution**, **intelligence**, and **indicator** were introduced as separate layers within the data model. They allow the assignment of individual CTI entities to classifications defined within the meta model. In the next step, each of the standardized entity types were integrated into the model. To preserve the expressiveness of the underlying data formats, all available entity relations were also integrated into the model. Finally, the included entities were extended by available complex attributes. These were extracted from the corresponding entities within the underlying data formats. As a result, this provides an overview of the structuring possibilities of the data formats and uncovers possible structural weaknesses.

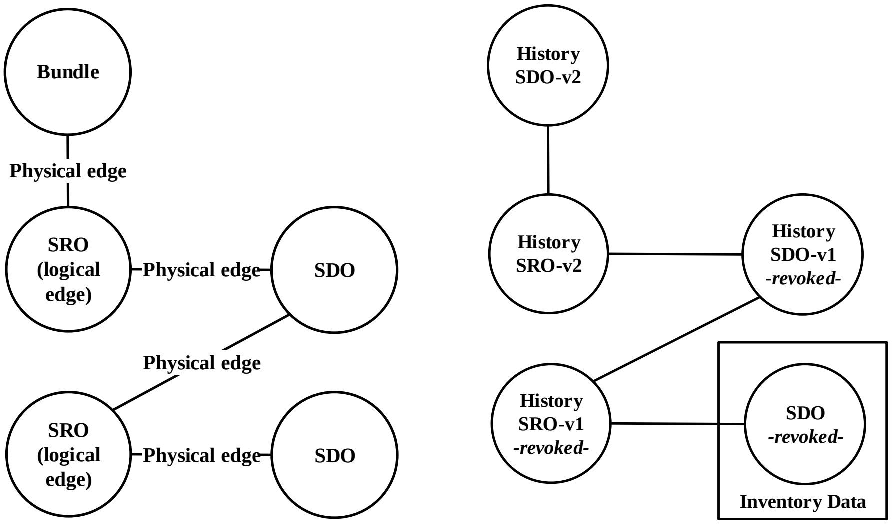
Contribution of P2:
With this work, a central definition for the essential attributes and properties of CTI data formats was created which allows the merging of data formats on the data type level. In addition, a set of rules was defined that enables existing notations and entity types to be standardized. These tools allowed the state-of-the-art data formats to be converted into a unified CTI data format. This allowed the functionality of the approach to be demonstrated and contributed to the identification of possible weaknesses of the underlying data formats.

4.3 Integration of human beings in CTI processing

After the previous section has provided the necessary fundamentals for the description of threat intelligence, the second focal area of the dissertation addresses the practical use of CTI information. Existing data formats offer a solid tool set for the structured description and storage of information. However, the included data is often complex and only available in data formats that primarily focus on the use case of machine processing. These are usually difficult to access for humans, particularly in the case of extensive data sets. However, since the human factor is an essential component of successful incident detection, analysis, and response, this section examines ways of efficiently integrating humans into the process. Specifically, two different perspectives are considered in this work. While Paper P3 examines the integration of domain experts into the analysis process, Paper P4 investigates possibilities for integrating employees without special security knowledge.

P3: Graph-based visual analytics for cyber threat intelligence

In the area of the integration of human beings in the handling of CTI, Paper P3 first looks at ways of involving security experts and thus directly addresses the research question RQ3 (see Section 2). For this purpose, the paper demonstrates how complex CTI data can be stored, processed, and made accessible to security experts. A major problem in the management of threat intelligence is the way in which the information is provided. Usually XML or JSON serializations are used which were primarily designed for machine processing. However, especially large amounts of data often lead to very low human readability and are therefore of limited use to security experts. This is problematic since security experts play a key role in incident detection and handling, and their success is critically dependent on the available information base. Therefore, the main objective of this paper is to provide experts with access to intelligence on threats, thus enabling the integration of expert knowledge into the analysis process. At the same time, interactions with CTI data lead to the additional requirement of ensuring the integrity of the underlying data and tracking changes. This is important to allow the data to be used as evidence after interactions with experts — for example, in subsequent court cases. In order to enable such an integrity-protected integration of expert knowledge, the knowledge-assisted visual analytics (KAVAS) approach for STIX was developed with this paper.

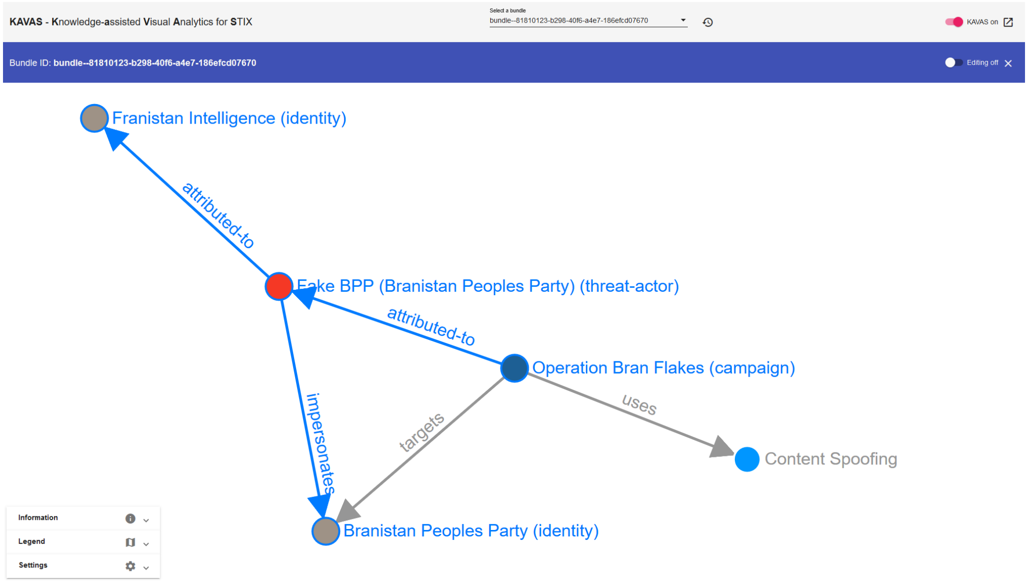


(a) Persistence mechanism (b) Integrity mechanism

Figure 8: KAVAS persistence mechanism based on Paper P3

KAVAS consists of two essential components. On the one hand, it allows to persist threat intelligence data within a graph database and to ensure the data integrity. On the other, hand it provides a visualization concept which allows interaction with the data and knowledge exchange with the system. The concept was finally evaluated by an analyst survey and several expert interviews. The basis for the data storage within KAVAS is a concept for integrity-secured persistence of threat intelligence data by means of a graph database. The data format STIX was chosen for the implementation because it is the most widely used data format in the field of CTI [22]. A STIX dataset essentially contains three object types — **Bundle**, STIX Relationship Objects (**SRO**), and STIX Domain Objects (**SDO**). While the bundle object encloses the data set, the user data is represented by SDOs and SROs. SDOs represent threat intelligence entities and SROs link individual SDOs.

In KAVAS, the relationships between the STIX entities are implemented in a graph database. These are illustrated in Figure 8a. SDOs, SROs, and the bundle object are shown as nodes in the graph. The objects in turn can be connected by physical edges in the graph. This type of data storage allows storing a history of changes in the graph without affecting the integrity of the initial dataset. More specifically, changes to the dataset can be represented by additional history nodes, as shown in Figure 8b. The base data is treated as inventory data, which is never changed. Actual changes to SDO or SRO objects are instead stored in history nodes and attached to the graph by adding auxiliary edges. This enables proving and reconstructing changes in the database retrospectively. In the second step of this work, a user interface for the STIX graph data was developed and prototypically implemented. Figure 9 shows the user interface, which provides an interactive graph that allows security experts direct access to the STIX data. From a methodological point of view, the user interface follows the knowledge-assisted visualization approach of Federico et al. [11]. The interface therefore pursues the goal of reflecting the four knowledge conversion processes of internalization, externalization, combination and collaboration, as shown below.



KAVAS - Knowledge-assisted Visual Analytics for STIX

Select a bundle: bundle--81810123-b298-40f6-a4e7-186efcd07670

KAVAS on

Bundle ID: bundle--81810123-b298-40f6-a4e7-186efcd07670

Editing off

Franistan Intelligence (identity)
attributed-to
Fake BPP (Branistan Peoples Party) (threat-actor)
attributed-to
Operation Bran Flakes (campaign)
uses
Content Spoofing
targets
impersonates
Branistan Peoples Party (identity)
Information
Legend
Settings

Figure 9: KAVAS user interface based on Paper P3

Internalization refers to the conversion of explicit knowledge from the database into tacit knowledge which corresponds to the user's understanding. To achieve internalization of the information provided, KAVAS offers a scalable representation of the STIX graph structure. It is based on a node-link diagram, allows interactive exploration of the dataset, and supports different filter options. **Externalization** refers to the transfer of expert knowledge into explicit knowledge within the data stock. This conversion process is achieved in KAVAS by providing the possibility to insert additional information, such as new nodes. The application also allows the editing of existing elements and enriching them with additional information. Because the explicit knowledge can be linked to existing data, **combination** is also available as a conversion process in KAVAS. Finally, the **collaboration** process is mapped by storing the information in one central graph database. This means that the externalized knowledge of experts is stored centrally, making it available to all other experts who may be working on the application. To verify the validity and usefulness of the results, the application prototype was evaluated in the final step of the work. For this purpose, a questionnaire-based user survey was first conducted among IT security analysts. It was demonstrated that the approach is of practical relevance and an appropriate tool for the analysis and processing of Threat Intelligence information. Subsequently, additional expert interviews were conducted to validate specific functionalities of the application. It was shown that the application clearly contributes to the understanding of threat intelligence for the user. Furthermore, it became clear that there is a great interest in such a tool in practice.

Contribution of P3:
In summary, this paper provided an interface for security experts to interact with threat intelligence in the STIX data format. At the same time, the integrity-protected data storage ensures a complete history of the database. The validity of the presented approach and the developed prototype was finally verified by an analyst survey and different expert interviews.

P4: Human-as-a-security-sensor for harvesting threat intelligence

Subsequent to the presented methodology for the integration of security experts in the analysis process, the second paper in this focal area deals with integration possibilities for individuals who have no security background. It examines how employees with different levels of knowledge can be integrated into the analysis and exchange process of CTI and thus it directly addresses the research question RQ4 (see Section 2).

A major difficulty in the analysis of attacks and security incidents is the acquisition of the necessary data. The data can either be obtained iteratively as part of system monitoring or extracted as highly detailed system snapshots for forensic processing. Although the data acquired in this way provides valuable information about system statuses and actions performed, the scope is usually limited to the respective system. As a result, only predefined events on specific systems can be recorded and detected; events outside these systems remain undetected. From a data structure perspective, this data contains mostly information from the indicator layer; contextual information from the attribution or intelligence layers (see Section 4.2) is not available. Contextual information, in contrast, can usually only be obtained from appropriate experts — for example, with the help of interfaces such as KAVAS (see Section 4.3).

To solve these problems in the data acquisition, this paper proposes a methodology for the integration of employees in the acquisition and analysis processes. The methodology allows persons who lack special IT security knowledge to act as human sensors and contribute additional information to the analysis process. This integration of humans opens up new possibilities. For example, information about security incidents can be made available that is either not visible at all or that becomes visible only at a later point in time when evaluating log data. In addition, humans as a data source can provide contextual information on the attribution and intelligence layers, thus expanding the spectrum of data acquisition. For example, attackers can be attributed, physical attacks can be detected, and the possible effects of a security incident can be evaluated in advance.

These insights have been utilized in this paper to develop a methodology that allows the information provided by individuals to be used in the analysis process. To achieve this, possible interfaces to Security Information and Event Management (SIEM) processes were first identified and a data model was developed to cover this additional data source. Subsequently, a CTI data structure was developed and implemented on the basis of the STIX data format. Finally, the concept was adapted into a prototypical application that allows the capture of human sensor data and its translation into the proposed CTI data structure.

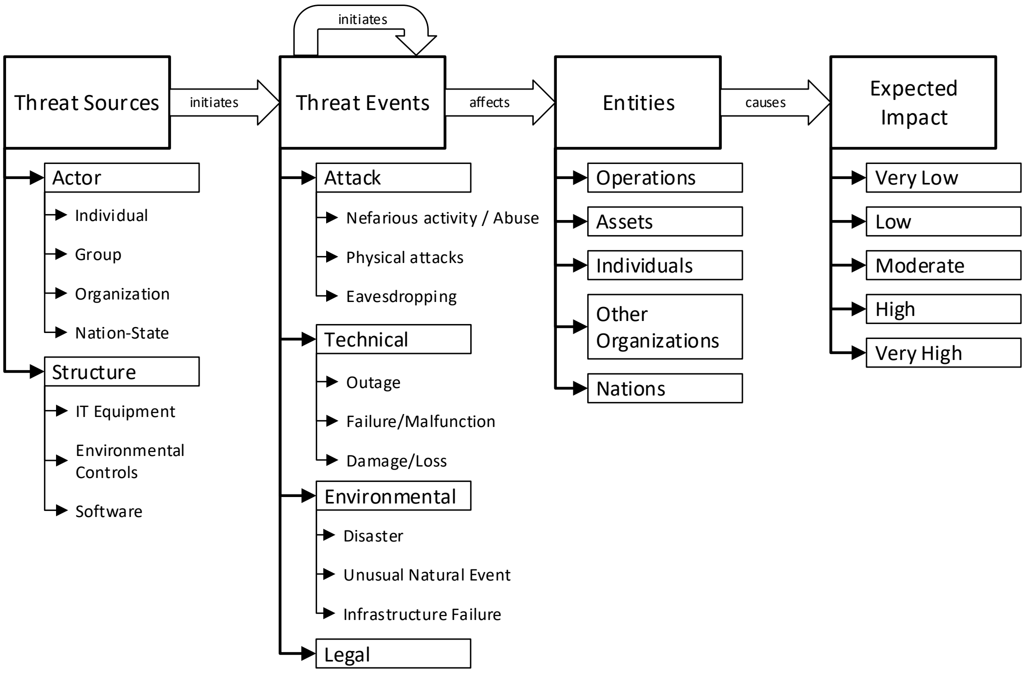
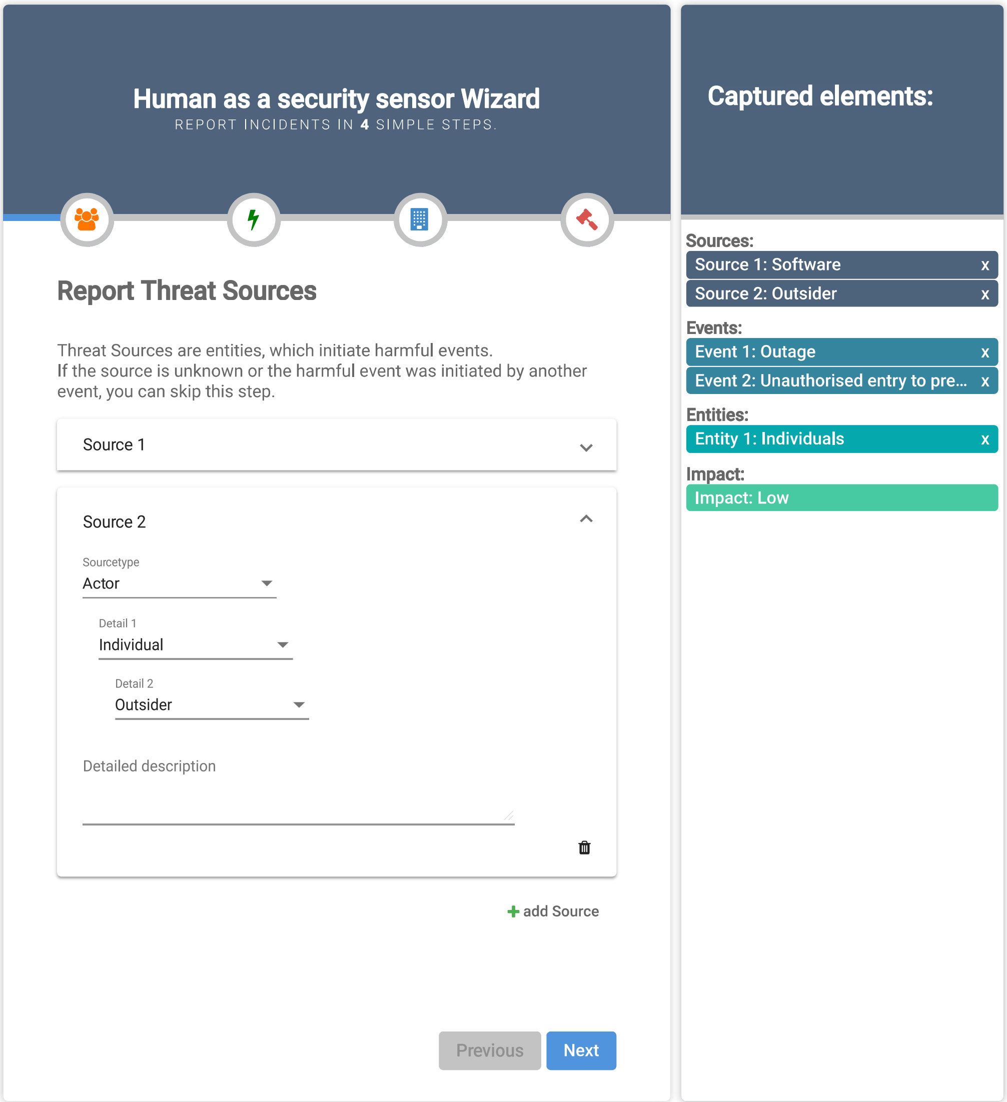


Figure 10: Incident model and taxonomy based on Paper P4

In the first step of the paper, SIEM systems were investigated for possible interfaces that allow the integration of human sensor data into the analysis flow. The push and pull approaches were identified. The push approach is based on findings of the individual which are actively reported to the system and supplement data from the automated data acquisition. The pull approach, on the other hand, describes a query from system to human to acquire additional or missing information following an analysis process. In the second step of this work, an incident model was developed which allows the acquisition of automated log data in combination with human sensor data. This is a central point of the present work, as only the application of this model makes it possible to capture the insights of individuals completely. Figure 10 gives an overview of the developed incident model, which is described in detail below. The incident model describes four major risk factors as process steps — **threat source**, **threat event**, **entity** and **impact**. These are derived from the works of Juliadotter and Choo [15] as well as from the National Institute of Standards and Technology (NIST) guidelines [14]. The risk factors also serve as essential activities within the model. Each activity was in turn subdivided into different areas to provide a detailed picture of the possible hazards. For the design of the individual activities, established guidelines and taxonomies of state actors as well as those from the scientific literature were used. Overall, the incident model represents incidents, starting with the definition of a threat source, through threat events which have occurred and entities which have been affected, to the impact that may be expected. It also allows the representation of traditional attacks as well as incidents that can be reported by human sensors, such as technical outages or disaster events.



Human as a security sensor Wizard

REPORT INCIDENTS IN 4 SIMPLE STEPS.

Report Threat Sources

Threat Sources are entities, which initiate harmful events.
If the source is unknown or the harmful event was initiated by another event, you can skip this step.

Source 1

Source 2

Sourcetype: Actor

Detail 1: Individual

Detail 2: Outsider

Detailed description

Captured elements:

Sources:
Source 1: Software
Source 2: Outsider

Events:
Event 1: Outage
Event 2: Unauthorised entry to pre...

Entities:
Entity 1: Individuals

Impact:
Impact: Low

Figure 11: Human-as-a-security-sensor wizard based on Paper P4

In the next step, the presented incident model was transferred into an extended, generic CTI data structure that allows representing both automatically collected information from log files and additional information from human sensors. The developed data structure is based on a comparison between the proposed incident model and the UPSIDE presented in Paper P1 (see Section 4.2). The comparison performed here ultimately yielded two results. First, it was possible to determine which aspects of human sensor messages are already describable. Second, it was possible to determine components that could not be represented within CTI data structures. Based on these findings, necessary extensions for the representation of human sensor data were identified, developed, and described within the generic data format. This, for example, includes elements for the representation of technical, environmental, and legal events, and thus supplements the representation possibilities of traditional attacks. The resulting generic data format was subsequently transformed into a tangible extension for the state-of-the-art CTI data format STIX to illustrate the practicability of the concept. In this process, the components of the generic data format were completely adopted through additional object types as well as extensions for the underlying threat definitions. The resulting extension to the STIX data format was additionally published with this work.

In the last step of this paper, the findings were prototypically implemented in the form of a reporting interface for human sensors, as shown in Figure 11. In the reporting process, employees are guided by a wizard that maps the four process steps of the incident model and queries the respective components of the incident model. To handle complex security incidents, the wizard also allows capturing several elements for each process step. Overall, the prototype allows information to be collected from the employee and recorded in the previously defined CTI data format for further processing in a SIEM analysis process. At the conclusion of the work, the actual benefit of the provided human sensor information was evaluated using the example of different use cases within a case study.

Contribution of P4:
In summary, this work showed that employees can be integrated into the SIEM analysis process as an additional data source. With the development of the threat model, it was possible to establish an extended CTI data structure that allows the findings of employees to be recorded in a structured manner. By addressing the corresponding interfaces, an additional integration possibility in SIEM data structures was demonstrated. As a result, better results in the detection of security incidents can be achieved and higher-quality incident data can be generated.

4.4 Overcoming obstacles in CTI exchange

The first two focal areas of this dissertation studied data structures for the representation of CTI and approaches for the integration of humans in the acquisition and analysis processes. Building on this work, the last focal area sought to examine how barriers to the exchange of CTI could be removed, thus creating a sustainable basis for information exchange. To achieve this, Paper P5 first examined how a SIEM analysis process could be designed in accordance with the legal requirements. Subsequently, Paper P6 explored possibilities for creating an incentive-based CTI exchange platform in compliance with the legal requirements.

P5: Towards GDPR-compliant data processing in modern SIEM systems

In order to remove possible obstacles in the exchange of CTI, Paper P5 first discusses the applicable legislation. A concept for SIEM systems was developed which allows the processing of threat intelligence as well as the fulfillment of legal reporting obligations as per the GDPR. For this purpose, the central SIEM processes were examined individually, a concept for legally compliant system was presented, and then it was evaluated technically and legally. The architecture of the research project DINGfest [16] served as the basis for this concept. Legal requirements regarding data protection can have a significant impact on the analysis and processing of information and thus also on work concerning threat intelligence. For example, the European GDPR [10] introduced in 2016 and the German Federal Data Protection Act [8] introduced in 2017 define various regulations and restrictions regarding the handling of personal data. At the same time, the collection and analysis of data within SIEM systems fall within the scope of privacy legislation, as it is usually not possible to fully prevent personal data from being processed.

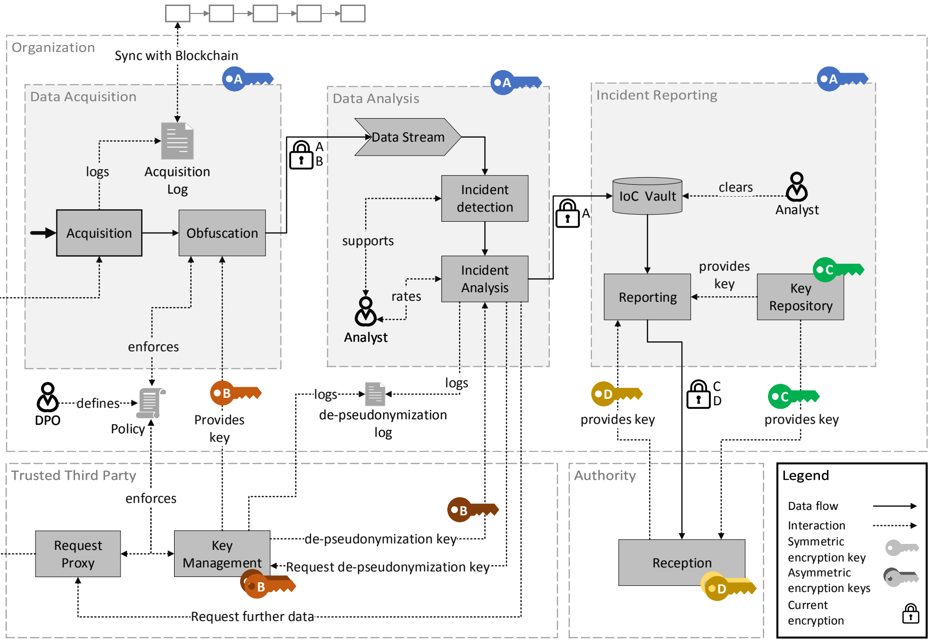


Figure 12: GDPR-compliant SIEM architecture based on Paper P5

Nevertheless, the aforementioned legislations allow the processing of personal data in combination with privacy protection procedures, such as encryption or pseudonymization. However, the use of such methods also creates new problems. Insufficiently protected data may lead to the risk of legal violations and excessively strong protection may lead to analyses which provide insufficient results. Hence, there is a tradeoff between the legally compliant protection of data and the usability of the results. In order to address this problem, a concept for SIEM systems was presented in this work which allows an analysis of security incidents while still being compliant with the legislation. As a starting point, the work focused on the essential questions that must be answered in order to develop such a system. First, it must be clarified what level of protection is necessary for the processed data to ensure conformity with the GDPR. This leads to the second question of whether security incidents can be detected on protected data sets and how this affects detection rates. Furthermore, it must be clarified under which circumstances and in which way protected data can and may be disclosed and reported according to the reporting obligations. To address these questions, the paper provides a comprehensive process model for legally compliant data processing within SIEM systems in combination with a procedure for the data analysis of protected data. Figure 12 shows the developed concept which combines both technical and organizational measures to enable a legally compliant analysis and the reporting of security incidents.

The concept divides the analysis functions within SIEM systems into three areas: data acquisition, data analysis, and reporting. The data within the areas is encrypted with key A to minimize the risk of data leakage to the external actors. In the first area, data is recorded in a structured form and the process is logged with integrity protection. The integrity of the protocol is ensured by storing checksums on the blockchain in accordance with Putz et al. [20]. In addition, the data is protected with the B key, which is provided by a trusted third party (TTP). This ensures, that no data leaves the collection process. A data protection officer (DPO) defines a policy that ensures the proper environment for the encryption and protection of the data. The analyses are performed using forensic fingerprint calculation as per Dewald [9] on the encrypted data. Only when a security incident is detected can the decryption key be requested from the TTP according to the data protection policy. Decrypted incident information can then be converted into a structured data format and persisted in long-term storage for incident information. From there, the data can be prepared and reported to the corresponding authority in a pseudonymized manner. The concept presented was evaluated twofold as part of this paper. First, a technical evaluation was carried out which shows that analyses on pseudonymized data lead to similar results to those on raw data, regardless of the reduced data pool. Second, a legal evaluation has shown that the proposed concept fulfils the necessary balance between freedom rights and individual protection interests, and thus follows the directives of the GDPR.

Contribution of P5:
In summary, this paper presented a technical and legally evaluated blueprint for GDPR-compliant data processing within SIEM systems. The entire processing cycle, starting with the collection of raw data via the analysis and reporting of structured CTI, was considered. The results showed that it is possible to simultaneously process information within the SIEM system as well as provide GDPR-compliant data protection and incident reporting.

P6: Decentralized Incentives for Threat Intelligence Reporting and Exchange

Papers P1-P5 established the foundations for the acquisition, representation, processing, and reporting of threat intelligence. Paper P6 addresses the exchange process and the general conditions that need to be met for the reporting. Specifically, a decentralized platform was created that allows the reporting of security incidents in an integrity-protected, non-repudiable form and provides incentives for the exchange of threat intelligence.

In the area of exchanging and reporting threat intelligence, companies are faced with various areas of tension. The legislation in various countries imposes reporting obligations for IT security incidents, especially in the area of critical infrastructures. In addition to the costs of detection and information processing, reporting obligations also lead to various requirements that must be fulfilled within the scope of a report. Since reporting obligations usually serve to protect the infrastructure of the respective community, there are increased requirements for the availability of the associated reporting infrastructure. Most reporting obligations are also accompanied by sanctions such as fines. It is also necessary to be able to prove that a report has been sent and that its contents are correct. This can be directly translated into specific requirements for the integrity and non-repudiation of reports. In addition to mandatory reports, an exchange of threat intelligence can also take place on a voluntary basis. This could strengthen the knowledge base of companies and thus increase the level of protection. At the same time, an exchange is also associated with costs and risks, such as unwanted information leaks. This could lead to companies either not actively participating in the exchange or not participating at all. For this reason, it is also important to provide incentives for active participation in an exchange.

With Paper 6, a decentralized ecosystem based on blockchain technology was developed which aims to address the problems discussed above. The system provides functionality for reporting security incidents and exchanging threat intelligence. During the development of the system, both the specific requirements for a reporting process and possibilities for creating incentives within the framework of a voluntary exchange were taken into account. The ecosystem designed for this purpose is shown in Figure 13. It provides an overview of the system, which is explained in detail below.

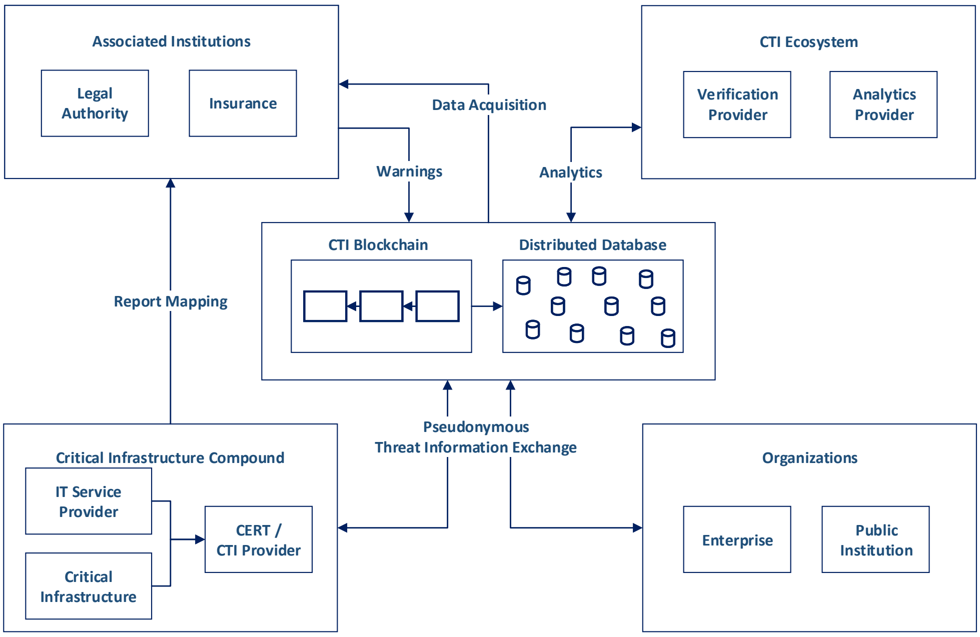
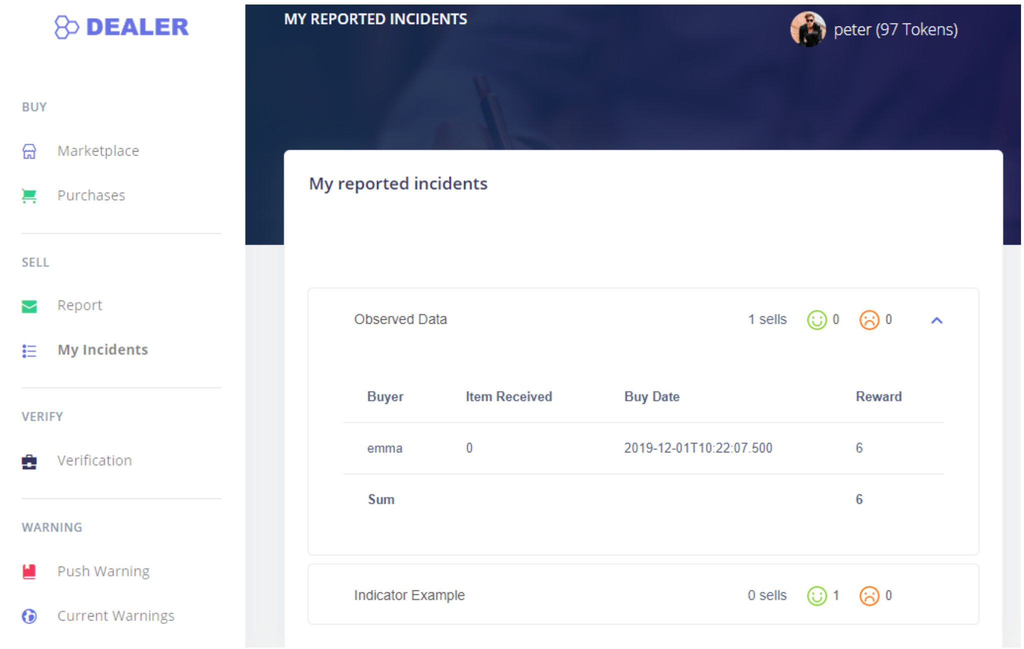


Figure 13: Decentralized ecosystem based on Paper P6

The center of this system consists of a CTI blockchain and a distributed database. While the actual data storage is handled by the distributed database, meta data for all processes are stored on the blockchain in an integrity-proof manner. The system also takes four main groups of participants into account. These include **critical infrastructure compounds** and **organizations** as active participants in exchange and reporting, as well as **associated institutions** as recipients of reports and participants in the **CTI ecosystem** as service providers. The system thus supports two main use cases: incident report and threat intelligence exchange. As part of a report, critical infrastructure associations can report information on security incidents to associated institutions, such as responsible authorities. Since the metadata for reports is stored on the blockchain, it is possible to check both the reporting activity and the integrity of the report content at any time.



My reported incidents

Observed Data 1 sells ☺ 0 ☹ 0

Buyer	Item Received	Buy Date	Reward
emma	0	2019-12-01T10:22:07.500	6
Sum			6

Indicator Example 0 sells ☺ 1 ☹ 0

Figure 14: DEALER user interface based on Paper P6

The platform also allows organizations to exchange information on security incidents with each other. For this purpose, a decentralized marketplace was developed that allows CTI to be offered for sale and thus provides a financial incentive for CTI exchange. To ensure the data quality of the information exchanged, it is also stipulated that the offers provided are checked by the associated CTI ecosystem. In addition to the separate use cases of reporting and exchange, the platform also allows them to be linked. This allows the CTI intended for a report to be verified and offered for sale on the internal market at the same time, thus allowing synergy effects to be harnessed. A major focus of the work was also to ensure a fair exchange between the individual participants. For this purpose, possible use cases and fraud possibilities were analyzed and addressed by appropriate processes within the platform. In the second step of the work, the Decentralized Incentives for Threat Intelligence Reporting and Exchange platform (DEALER) was implemented as proof of concept of the designed system. The technical basis of the system comprises

EOS[6] as the public blockchain, IPFS[7] as the data storage, and a NodeJs[8] application as the decentralized user interface. Figure 14 shows the user interface of the DEALER platform, which is briefly described below.

The application provides three main functional areas. The **BUY** area allows organizations to purchase incident information on the marketplace, thus implementing the incentive-based exchange of the platform. The **SELL** area allows critical infrastructures or organizations to report information on security incidents, make it available for sale, or combine both actions. This area is used to implement previously identified reporting requirements. Finally, the **VERIFY** area is available to participants of the CTI ecosystem. Here, offered data sets can be evaluated and thus the quality of the data inventory on the platform can be ensured. The work was concluded with an extensive evaluation of the platform. This included an assessment of the costs associated with the use of a public block chain, as well as the fulfilment of reporting requirements and possible security issues.

Contribution of P6:
This paper introduced a decentralized platform for CTI reporting and exchange. On the one hand, this created the necessary basis for fulfilling the obligation to report security incidents in an integrity-protected and non-repudiable form. On the other hand, the implementation of the platform on the basis of blockchain technology allowed the creation of trustless financial incentives within the exchange process. The results of this work provided the basis for exchanging structured CTI with third parties and making this information available as a report.

4.5 Complementary publications

In addition to Papers P1-P6, further research has been done in connection with this dissertation. Although the additional papers A1-A3 are not directly included in the present dissertation, they have influenced the results presented here in various ways and are therefore briefly presented below. Table 2 gives an overview of these papers. It shows the full citation of the publication, its submission status, and the publication type — (C) for conference and (J) for journal.

With **Paper A1**, a basic architecture for modern SIEM systems was developed and presented at the S*icherheit 2018* conference [16]. This work served as the basis for the architecture proposed in Paper P5 which was extended by different concepts to achieve a GDPR-compliant SIEM system.

In **Paper A2**, a blockchain-based infrastructure for auditing log data was developed and published in the journal *Computers & Security* [20]. The paper proposes a mechanism to allow the integrity-protected auditing of log data. Due to the implementation of the concept with the help of blockchain technology, a trustless operation of the auditing is

[6] https://eos.io/

[7] https://ipfs.io/

[8] https://nodejs.org/en/

No.	Publication	Status	Type
A1	Menges, F., Böhm, F., Vielberth, M., Puchta, A., Taubmann, B., Rakotondravony, N., and Latzo T.: Introducing DINGfest: An architecture for next generation SIEM systems. *Sicherheit 2018*: 257-260 (2018)	published	C
A2	Putz, B., Menges, F., and Pernul, G. A secure and auditable logging infrastructure based on a permissioned blockchain. *Computers & Security 87* (2019)	published	J
A3	Schlette, D., Menges, F., Baumer, T., and Pernul, G. Security Enumerations for Cyber-Physical Systems. *Data and Applications Security and Privacy XXXIV, DBSEC* (2020)	accepted for publication	C

Table 2: Overview of complementary research papers

possible. The results of this work have also been incorporated into the outcome of Paper P5 and allow for a provably integrity-assured data acquisition within the SIEM system.

Furthermore, an enumeration for cyber physical systems was developed in **Paper A3**. This work was accepted at the *DBSEC 2020* conference and is expected to be presented and published in 2020. The work addresses the problem that common threat intelligence methods can represent IT systems but the available tools are unsuitable for the representation of cyber physical systems. As a result of this work, an extension for CTI data formats for the representation of cyber physical systems was presented. This extension builds directly on the structural basics of the Papers P1 and P2, and extends them using components for the representation of cyber physical systems.

5 Conclusion and Future Work

The field of CTI is still a relatively young discipline with a limited research corpus. Therefore, the goal of this dissertation was to contribute to the development of the foundations for the use of CTI in general and the exchange of CTI in particular. In order to achieve this goal, three main foci were set in the work. First, methodological foundations for CTI data structures and data formats were developed. In the second step, possibilities for involving individuals with different levels of knowledge in the practical use of CTI data structures were investigated. Finally, possible obstacles for the use and exchange of CTI were identified and solutions were proposed to mitigate their impact. Taken together, the goal of the work was to create the methodological and technical basis for sustainable use and exchange of threat intelligence.

In the course of this dissertation, several essential contributions to the research area of threat intelligence sharing were created. First, the CTI data structures, and data formats were comprehensively analyzed. This resulted in a process model for capturing structured incident information, a generic object model for representing threat intelligence data structures and a meta model for describing threat intelligence data types. With the development of these models, this dissertation provides fundamental and comprehensive tools for the description of threat intelligence. In this context, process models were developed for the analysis, comparison, and standardization of threat intelligence data formats. These represent an additional tool for a uniform exchange between parties with heterogeneous data structures.

Besides these methodological foundations, this dissertation also examined possibilities for the involvement of humans in threat intelligence work. The first result in this area was the interactive integration of experts into the analysis process of CTI data. This opens up the possibility to validate analysis results and enrich them with contextual knowledge. In addition, a procedure for the integration of people without a special IT security background into the data acquisition could be conceptualized and realized. Employees act as so-called human sensors and extend the data acquisition as an additional data source. This makes it possible to extend the database with additional contextual information and to provide information which would potentially not be provided by log data.

Finally, the third focal area of the dissertation covers potential barriers to the processing and exchange of threat intelligence and provides solutions to mitigate these barriers. First, the impact of legal requirements on the processing of personal data and on the reporting of security incidents were examined. Subsequently, a SIEM system was developed that allows the processing of protected personal data as well as the reporting of detected security incidents in accordance with legal requirements. In a second step, requirements and threats resulting from the exchange of CTI were examined in detail and addressed by a decentralized sharing platform. The platform allows essential reporting requirements to be met, such as integrity assurance and non-repudiation. In addition, the platform also enables the creation of incentives for the exchange of CTI, thus offering a counterweight to possible risks associated with the exchange of information.

In summary, this dissertation provides the methodological foundations for structuring threat intelligence, possibilities for integrating individuals into the analysis process, and the basis for a legally compliant and incentive-based exchange of CTI. This covers the entire cyber threat intelligence lifecycle, right from the collection and processing of threat intelligence, through expert analysis of the data to the incentive-based reporting and review of identified security incidents, as outlined in Section 1.

In addition to the immediate results, various starting points for future developments and research were identified in the course of the dissertation. It has become apparent, for example, that the existing CTI data formats do not adequately reflect certain information, such as impact assessments or countermeasures to security incidents. Although basic information can be represented in most data formats, there is often considerable potential for improvement in the respective level of detail. In addition, this work has provided a model for the standardization of CTI data structures and thus represents an advance toward a unified standard. However, the long-term success of such a standardization depends largely on its implementation in practice and thus on active use.

The studies concerning the integration of humans into the analysis process provided different starting points for future research. The expert interviews conducted have shown that the approaches developed in this work are promising but that there is still a great need for further research in this area. This includes, for example, in-depth research into the possibilities for extensive cooperative work on security incidents as well as the consolidation of structured threat intelligence. The work on the integration of employees as human sensors has also raised new questions in addition to the actual results of the research. On the one hand, these include possibilities for creating incentives for active participation on the sensor platform. On the other hand, aspects of data protection law and implications of possible mutual accusations of employees on the human-as-a-security-sensor platform can also serve as starting points for future work. Moreover, with the work on the implementation of a GDPR-compliant SIEM system and the incentive-based platform for the reporting of security incidents, a possible practical implementation of reporting obligations and voluntary exchange was shown. However, the actual acceptance of the procedures among employees in practice still needs to be proven.

Part II

Research Papers

1 A comparative analysis of incident reporting formats

Current status:	Published	
Journal:	Computers & Security, Volume 73, March 2018	
Date of acceptance:	27 October 2017	
Full citation:	Florian Menges and Günther Pernul. A comparative analysis of incident reporting formats *Computers and Security 73, 87–101 (2018).*	
Authors contributions:	Florian Menges	90%
	Günther Pernul	10%

Journal Description: Computers & Security is the most respected technical journal in the IT security field. With its high-profile editorial board and informative regular features and columns, the journal is essential reading for IT security professionals around the world.

COMPUTERS & SECURITY 73 (2018) 87–101

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

A comparative analysis of incident reporting formats

Florian Menges [*], *Günther Pernul*

Department of Information Systems, University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany

ARTICLE INFO

Article history:
Received 21 April 2017
Received in revised form 11 October 2017
Accepted 27 October 2017
Available online 6 November 2017

Keywords:
Incident reporting
Incident management
Incident response
Reporting formats
STIX
IODEF
IODEF-SCI
VERIS
X-ARF

ABSTRACT

Over the past few years, the number of attacks against IT systems and the resulting incidents has steadily increased. To protect against these attacks, joint approaches, which include the sharing of incident information, are increasingly gaining in importance. Several incident reporting formats build the basis for information sharing. However, it is often not clear how to design the underlying processes and which formats would fit the specific use cases. To close this gap, we have introduced an incident reporting process model and the generic model UPSIDE for basic incident reporting requirements. Subsequently, we have identified state-of-the-art incident reporting formats and used the introduced models to conduct a comparative analysis of these formats. This analysis shows the strengths and weaknesses of the evaluated formats and identifies the use cases for which they are suitable.

1. Introduction

The number and complexity of IT systems as well as the number of potential vulnerabilities compromising these systems have been steadily growing over the last years. This growth comes along with a likewise increasing number of potential threats to the systems. These threats range from autonomous self-replicating malwares with various obfuscation characteristics, which are not only restricted to affecting software but also infect hardware components with highly sophisticated targeted attacks. Altogether, this leads to a noticeable increase in successful cyberattacks resulting in both economic damage and loss of data (McAffee Corporation, 2016). Moreover, the implications that arise from such threats are not necessarily restricted to the IT landscape; they can also reach entities within the physical area and therefore can, in a worst-case scenario, even influence the critical infrastructure of a region, country or the whole society.

In the last decade, there have been significant research and development efforts in the area of threat intelligence. These include activities to mitigate damage in case of already occurred harm. However, it can be observed that traditional isolated defense approaches only provide security under certain conditions and therefore mostly do not meet the requirements to protect systems and infrastructures against today's threat landscape (Symantec Corporation, 2016). Since this can mostly be attributed to an incomplete information basis, one possible approach for improving the current situation and thus the overall security of systems is the sharing of threat infor-

[*] *Corresponding author.*
E-mail address: florian.menges@ur.de (F. Menges).

https://doi.org/10.1016/j.cose.2017.10.009
0167-4048/© 2017 Elsevier Ltd. All rights reserved.

mation along with cooperation between victims and authorities (Johnson, 2003). Cooperative approaches can substantially strengthen the information basis. Such approaches accordingly allow the improvement of threat detection and mitigation of current as well as future attacks due to the enhanced knowledge of every single participant. Therefore, it can be presumed that threat exchange technologies will prospectively develop into one of the key cyber threat defense technologies within companies.

Such information exchange has recently been stipulated by law for critical infrastructure operators in various countries such as the European Union (European Commission, 2016), Germany (Deutscher Bundestag, 2015), and the USA (Congress of the United States of America, 2014). The exchange of information itself can take place between companies, CERTs, and governmental institutions.

The key element within threat intelligence-sharing techniques is the utilized data formats because they pre-define which information would be shared. Additionally, the used data format implicitly defines requirements for the information density of the respective data elements. In the area of data exchange, the formats for an automated exchange and the processing of information about threats and incidents are widely anticipated (SANS Institute, 2015).

Even though there are different approaches to automated threat intelligence-sharing, the body of literature is still quite limited. To the best of our knowledge, no comprehensive analysis of data formats in use gathering all significant aspects was performed in the past. In particular, the current versions of the two most important data formats, namely STIX and IODEF, have not been adequately covered within the academic literature yet. Owing to the increasing importance of incident reporting, we believe that a thorough analysis of all relevant formats is an essential factor for future research.

Against this background, the remainder of this paper is organized as follows. In Section 2, we provide an overview of the related work in the area of incident reporting formats. In Section 3, we propose a general model for an incident reporting process and incident reporting formats. Based on this, we provide a comprehensive overview of contemporary available incident reporting formats in Section 4. This is followed by the development of criteria for the comparison of reporting formats in Section 5. In Section 6, we provide an evaluation of the identified reporting formats that aims to support the decision processes and the selection of an appropriate exchange format. The paper is concluded in Section 7.

2. Related work

Even though a lot of work has been done in the area of incident management and incident response in recent years, only a handful of researchers have focused on the data structures and processes for the exchange of security incident information. To get a detailed picture of available work in this area, we conducted a literature review on incident reporting and reporting formats. Next, we examined the available information for each of the identified reporting formats.

ENISA (Dandurand et al., 2014; ENISA, 2013) and Kampanakis (2014) provide descriptive overviews of the formats that can be used in an incident reporting process. Although most of these formats do not show any practical or scientific relevance nowadays, they give a broad overview of the different available format approaches. By contrast, our work identifies currently relevant formats and focuses on these while still considering less relevant formats.

Besides these descriptive works, there are some comparative approaches. Fenz et al. (2008) analyze the semantic potential of exchange standards. They provide a brief comparison based on semantic usability, information complexity, and distribution. They also provide a synopsis of the strengths and weaknesses for each of the examined formats. This work from 2008 mostly covers formats that have little relevance today. However, it provides a good starting point for comparing the current standards. Steinberger et al. (2015) analyze different exchange standards and introduce several comparative criteria. Their work provides valuable suggestions for comparing the reporting formats, even though most of these standards have little relevance today and the work does not specifically focus on reporting formats. Asgarli and Burger (2016) propose an approach that applies a quantitative comparison of the reporting formats by comparing the amount of classes and properties available within a format. This approach was applied to the first versions of STIX and IODEF. Although this type of analysis only covers specific aspects for the reporting formats, it provides additional informative value. In this work, we also apply a comparative approach by focusing on the currently relevant formats. Therefore, we evaluate the criteria introduced by related work and apply them, if appropriate. Additionally, our comparison is based on a combination of qualitative and quantitative criteria.

Furthermore, a lot of work has been done in the area of describing incidents and their components (e.g. Howard and Longstaff, 1998; Cichonski et al., 2012; Blackwell, 2010). However, neither possible relationships within incident reporting formats nor necessary adjustments for this type of use have been covered. Similarly, there have been different suggestions for representing an incident detection process. Such suggestions mainly originate from incident management and response (e.g. Prosise and Kevin, 2003; Deniz and Celikoglu, 2011; Munteanu et al., 2014) as well as computer forensics (e.g. Freiling and Schwittay, 2007; Ciardhuáin, 2004; Yusoff et al., 2011). This work considers many different aspects of the incident detection process. However, specific characteristics that enable a detection process to properly prepare contents for any information exchange are not covered. Therefore, we consider these characteristics.

To sum up, we want to close gaps within the related work by proposing a general process and model for incident reporting, followed by an identification of today's state-of-the-art incident reporting formats. Thus, we analyze the properties, strengths, and weaknesses of the identified formats. This analysis is based on our general model, the methods derived from the literature, and the newly proposed methods.

3. A general model for incident reporting

As the foundation for building a model for incident reporting formats, the basic entities are determined in this context. Sub-

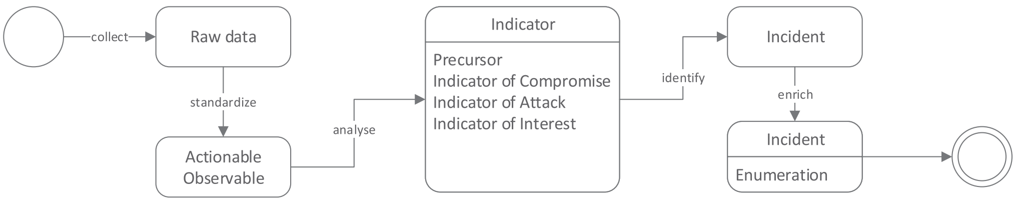


Fig. 1 – Incident detection process.

sequently, we take a closer look at the incident detection process and data structures within this process. Based on the determined entities and data structures, we establish a generic model for an incident reporting format.

3.1. *Basic elements of an incident reporting process*

To form a basis for the development of a generic incident reporting model, it is necessary to start with a closer look at the corresponding terms and definitions within the incident reporting process. Based on Howard and Longstaff (1998) and Cichonski et al. (2012), the terms event, attack, indicator, and incident in the context of the security incident reporting process will be explained below. Following their classification, an *event* is defined as observable occurrence within a system or network that can cause a discrete change of state or in the status of a system or device. The term *attack* refers to one or more such events on a computer or network; it results in something that is not authorized to happen. Attack consequences in turn can either lead to an incident, which is defined as the violation or the imminent threat of violation of computer security policies, acceptable use policies, or standard security practices, or at least, to an *indicator*. An indicator is defined as a sign that signals the possible occurrence of an incident. Accordingly, an *incident* represents an actual occurrence, while an indicator only represents a possible occurrence.

3.2. *Data structures in the incident detection process*

Based on the above-mentioned definitions, the incident detection process will be closely examined in this section. For this purpose, the necessary steps within this process, starting with collecting system events through to the detection of an attack or incident, will be considered. In particular, the transitions between the relevant data structures will be illustrated to create a better understanding of incident reporting.

The detection of security incidents usually begins with the collection of events, which have to be obtained as *raw data* from various heterogeneous sources such as log data from the analyzed systems, the capture of the network traffic and the states of systems, processes or devices. Hence, the data pool for analysis consists of highly heterogeneous data structures with varying semantics. At this point, it is necessary to normalize the obtained data to enable further detection and processing. This can be achieved by using a structured mark-up format for the representation of events as *actionable observables*. The examples of standardization approaches are the Cyber Observable Expression language CybOX[1] and applications such as Logstash[2] that enable the transfer of log data into a structured format.

After establishing a standardized data pool, it can be analyzed to determine the indicators of possible security-relevant occurrences from one or multiple events. The indicators can be differentiated by indications for possible events in future (*precursors*), indications for a possible compromise of the current system (*indicators of compromise*), indications for specific attacks on the system (*indicators of attack*), and other security-related information (*indicators of interest*) (IBM Corporation, 2015; Pirc, 2016).

Subsequently, specific incident entities can be identified by combining the previously detected indicators with detailed analyses. Depending on the data format, incidents can be enriched by additional information such as information about the attacker or the affected system. In addition to the core components of an incident detection process, *enumerations* can be used to arrive at a common understanding of threats and the involved components. Such enumerations represent unique definitions of systems or patterns that can be attributed to the indicators or incidents (MITRE Corporation, 2011). After the identification of a known attack pattern resulting in an incident, the pattern can be matched with an enumeration online available such as Common Attack Pattern Enumeration and Classification (CAPEC)[3] and attributed to the incident. This prevents misunderstanding within the detection process and can lead to a faster incident handling. Fig. 1 shows the components and event types within the incident detection process.

3.3. *A generic model for incident reporting*

After determining the basic elements and a generic process for incident detection, the generic representation of an incident will be established in this section. The basis for this model is the incident definitions and the incident detection process given above. Additionally, it is necessary to take a closer look at the actual components of an incident, which can be represented as a causal chain describing the root causes of an incident along with the resulting effects, as shown by ENISA (2010). A causal chain originates from an *attacker* or a malefactor that can have different aims including financial profit. An attacker performs attacks by using *methods and tools* that exploit the given *vulnerabilities* in the target system. These attacks are represented as *actions*. Finally, the *results* of an attack are represented by a victimized object as well as the attack's unauthorized result. Moreover, the casual chain can be extended by adding a defensive category (Blackwell, 2010). All the

[1] https://cyboxproject.github.io/
[2] https://www.elastic.co/products/logstash
[3] https://capec.mitre.org/

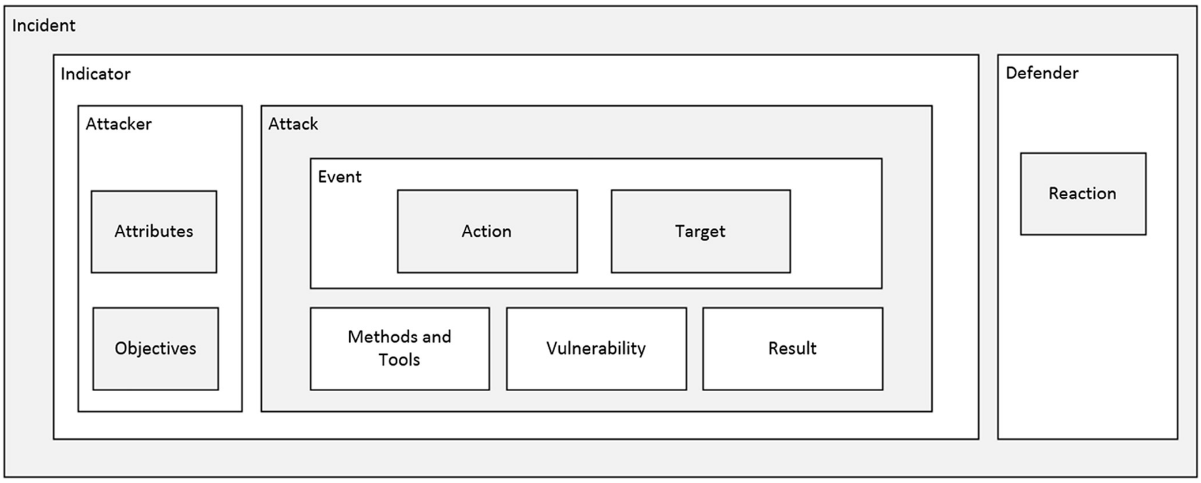


Fig. 2 – UPSIDE (a Universal Pattern for Structured InciDent Exchange).

components discussed are essential for a general incident reporting format. We have chosen an interleaved representation (see Fig. 2), which we call *UPSIDE* (a Universal Pattern for Structured InciDent Exchange). We also use this model to create the evaluation criteria for incident reporting formats in Section 5.

The base structure of UPSIDE is represented by an incident, in combination with an indicator which builds the frame for further components. An indicator can be characterized as a combination of information about the attacker and the attacks performed. The attacker is basically defined by its base *attributes* and specific *objectives* pursued. An *attack* in turn can be specified as a combination of the observed events on the one hand and the identified methods that use vulnerabilities to achieve a certain result on the other hand. Events represent a relationship between an action and an affected *target* in this context, both of which represent the base structure for any detection process. Finally, the section *defender* represents the counterpart for an indicator, which performs *reactions* based on the detection results with the objective of preventing or mitigating damage.

Fig. 2 illustrates that an incident is built out of indicators, which in turn refer to specific attacks. It is shown that an attack is always based on events that are enabled by methods and tools that exploit vulnerabilities with a certain result. The attacker entity is represented with its attributes and objectives. Moreover, the defender entity (suggested by Blackwell (2010)) is represented with its corresponding reactions to an attack. Consequently, this model covers all elements from the incident detection process through the active attacker and defensive measures.

4. Incident reporting formats

As argued above, some studies on the formats for threat intelligence-sharing can already be found in the literature. However, to the best of our knowledge, there is no publication available that covers all relevant reporting formats or examines important formats in depth. To close this gap, we give a contemporary overview of today's incident reporting format landscape in this section. The data exchange formats given in the literature cover the formats for the exchange of raw system data, the formats for the exchange of structured event data, the formats for sharing vulnerability information, and formats that represent the full spectrum of incident information.

We focus on the formats for the representation of incidents and thereby define an incident data structure as the minimum requirement. Formats that do not fulfill this requirement, such as formats that represent only actionable observables, are excluded. Furthermore, we constrain the detailed illustrations to formats that have a certain relevance. These are STIX, IODEF, VERIS, and X-ARF. Additionally, we examine the two major releases of the most important formats STIX and IODEF separately. Since these have been published most recently, it can be assumed that their earlier versions still show a high degree of relevance till date.

Table 1 – Most relevant incident reporting formats.

Name	Release	Last Update	Issuer	License	Serialization	Validation	Latest Version
STIX v.1	2012	2017/03	MITRE Corp.	BSD license	XML	XML-schema	1.2.1
STIX v.2	2017	2017/03	Oasis CTI	Oasis Open	JSON	JSON-schema	2.0 CSD
IODEF v.1	2007	2016/11	IETF	IETF TLP	XML	XML-schema	IODEF-SCI
IODEF v.2	2016	2016/11	IETF	IETF TLP	XML	XML-schema	2.0
VERIS	2010	2017/02	Verizon Inc.	CC BY-SA 4.0	JSON	JSON-schema	1.3.1
X-ARF	2011	2016/10	abusix GmbH	Open source	YAML	JSON-schema	0.2

Table 1 gives a short overview of all formats that have been examined in this paper. This includes basic information about their first release and information about their latest updates as a first indicator of the sustainability of the formats. Moreover, any information about the format issuer in combination with the release license indicates possible dependencies on specific companies. In addition, information about the used serialization as well as validation languages is outlined to highlight the possible performance aspects of each format.

Starting from this overview, each of the formats will be introduced, followed by a detailed overview of its specific components. After the examination of the relevant formats, formats that are classified as less relevant will be discussed at the end of the section. This includes a brief introduction and makes the justification for their classification as less relevant.

4.1. *Major incident reporting formats*

4.1.1. *STIX: Structured Threat Information eXpression (version 1)*

STIX is an approach to standardize the sharing of information relating to cyber threat intelligence in a holistic way. By using a heavyweight XML format, in combination with an XML-schema validation, it provides extensive incident reporting capabilities. STIX represents the components in a hierarchical structure with pre-defined attributes and fields; it provides indicators of compromise, threat intelligence, corresponding countermeasures, and reporting capabilities for incidents. Moreover, it supports external references such as enumerations. Additionally, STIX defines each component with a confidence level to indicate the reliability of its findings. It acts together with its own sub-language that represents low-level system events called CybOX. With a share of 46%, STIX is the most commonly used format for incident reporting within and between companies. Therefore, it is entitled as the de facto standard for describing threat intelligence data (Sauerwein et al., 2017). Moreover, according to the project issuer, the STIX format definitions are already supported by more than 60 IT-security software products.

Components: The main STIX components describe an incident in combination with the detected *indicator* objects. The representation of the attacker is realized by a *threat actor* object. The attacks in turn are represented by using *TTP (tactics, techniques, and procedures)* on the attacker side, while the *exploit target* is on the victim side and *observable* objects represent the specific events within an attack. STIX also supports the representation of the defender's perspective by employing *course of action* objects to express the intended as well as realized defensive measures.

CybOX observables represent raw system **events** in an autonomous format. CybOX is integrated in the STIX data format and intended to be used in combination with STIX. However, this separation of the event and incident formats allows the use of alternative presentation forms for system events. The observables form the basis of all further STIX components and are intended to be flexible enough to represent all kinds of measured system events as well as more abstract patterns. Threat actors are characterizations of **attackers** or malefactors within systems. These include their motivation, **objectives**, and intended effects in the context of an attack as well as information about the sophistication level of the attacker. STIX also provides the possibility to represent historical information about threat actors as well as relating to them, thereby leading to the expression of entire attack campaigns. As a counterpart to threat actors, STIX describes **exploit targets** uncoupled observable entities as potential targets for an attack, which basically describe assets such as systems, software or networks. This structure also includes information about **vulnerabilities**, weaknesses and system configurations, each of which can be referenced to specific enumerations for distinct identification. STIX designates the enumerations provided by MITRE and third party enumerations. STIX also provides capabilities to capture **threat intelligence** both for the attacker and the defender. In this context, TTP entities provide the possibility to capture patterns, exploits or malware, resources such as persons, tools and infrastructure used by the attacker and the intended effect of the attack. Course of action provides capabilities to capture the **countermeasures** taken by the **defender** as well as information about the likely impact caused by the incident.

The structures that have been presented so far build the basis for the indicators in the STIX format which are created from the analysis of one or more observables. Once an indicator is detected, it is defined by establishing relations with the corresponding entities. Therefore, it can be interpreted as the connecting piece between STIX incidents and all further included entities (Barnum, 2014).

4.1.2. *STIX: Structured Threat Information eXpression (version 2)*

STIX2 is a dynamic approach for the standardization of cyber threat intelligence-sharing based on STIX. Using JSON serialization mechanisms, it provides data transfers with little overhead. STIX2 provides a dynamic approach for representing relationships between the entities by introducing a relationship object that can be used to connect any entities within the STIX framework. Additionally, CybOX as an independent language has been removed from the standard and merged into the STIX language as *observable* objects. Along with these substantial changes, there have also been changes to the core data model, as shown below.

Component changes: In addition to the *relationship* object, STIX2 introduces several further top-level elements such as *intrusion set*, *identity*, *attack pattern*, *malware*, *tool,* and *sighting*. STIX has already introduced the threat actor as a representation of an attacker, which in turn can be associated with attack campaigns. This relationship is extended in STIX2 by an intrusion set as well as by an identity.

An **intrusion set** object represents the behaviors and resources that can be attributed to threat actors and campaigns. This allows a more precise specification of attackers and their intentions. In addition, an **identity** object represents individuals, organizations, or groups, to which threat actors could be attributed. Moreover, the TTP object is replaced by the entities **attack pattern**, **malware**, and **tool,** which are included in STIX as the sub-elements of TTP. This referential change supports the dynamic approach for creating relations within STIX2, thus allowing the expression of additional coherences.

Moreover, a new object **sighting** is introduced. It represents an extension to observable objects that enables more dynamic referencing between the observables and any other top-level object. This, for example, enables the references between an indicator, specific attackers, and their specific attack methods at the same time, while every top-level element specifies its own references to specific observables within the specification of STIX (Wunder et al., 2017).

4.1.3. IODEF: Incident Object Description Exchange Format (version 1)

IODEF defines an approach for an attack-centric exchange of incident information. Using XML as the data representation as well as XML-schema for data validation, it prescribes a base structure for representing an incident. IODEF offers predefined attributes and fields along with free-text fields for additional non-structured information within the base incident structure. However, IODEF does not provide explicit definitions for the use of external references such as enumerations. Addressing this was one of the main intentions of introducing IODEF-SCI as an extension to IODEF. It maintains the base structure of IODEF and extends it by adding fields for external referencing such as an enumeration for system configurations. IODEF provides the components for the representations of actual occurrences within an incident including the used methodology and a representation for the impact assessment. According to the SANS Institute (2015), IODEF is used by 23% of those companies that share threat information and is practically relevant for threat intelligence-sharing platforms (Sauerwein et al., 2017).

Components: The main IODEF components for describing an incident are *event data*, *flow*, *method*, *assessment*, *additional data,* and *history*. The basic data structure for representing incidents is described by **event data**. It consists of base attributes such as event description and date. It also describes the actual occurrences by using the so-called **flow** objects, which represent the technical event details such as hosts, networks, services, or system states. **Method** basically describes the methodology used by an attacker to reach its goals by using a list of free-text fields in combination with optional URL fields. These URL fields can, for example, be used to reference information about the detected malware or a used vulnerability. Furthermore, with the use of the IODEF-SCI extension, these properties can be enriched by using enumerations for a unique identification of the respective entities. **Assessment** can be considered as a result of the incident and includes various types of information about the incident impact including the information about the confidence in the respective assessment. In the analogy to the method, the assessment can also be attributed to the incident and specific event data.

Moreover, IODEF provides the possibilities to express non-structured **additional data** for each of the defined entities. Finally, the **history** element provides information about the handling of previous incidents.

4.1.4. IODEF: Incident Object Description Exchange Format (version 2)

IODEF2 is an approach for the exchange of incident information; it introduces several extensions for the core model of IODEF. In addition to the attack information focused by IODEF, IODEF2 introduces data structures for indicators, attackers, and defensive measures. This changes the attack-centric view of IODEF to a more holistic approach. Furthermore, IODEF2 adopts the capabilities for the external referencing introduced in IODEF-SCI.

Components: IODEF2 extends IODEF by the entities *indicator*, *threat actor*, *campaign,* and *course of action*. The additionally introduced **indicator** implies a change in semantics toward IODEF. IODEF allows the expression of incidents that only provide the confidence levels for the respective impact assessment, while IODEF2 supports the differentiation between incidents and evidence for threats or incidents. The indicator elements are intended to be a direct sub-element of incidents. **Threat actor** as a representation for an attacker entity is introduced with a supplementary **campaign** object, thereby providing a representation for the attack series attributed to the threat actors. Additionally, both the history and event data items are extended by **course of action** elements. The assessment entity is extended by a mitigating factor; this represents defensive measures and information. All in all, these changes lead from an attack-centric model to a full-scale representation of incidents.

4.1.5. VERIS: The Vocabulary for Event Recording and Incident Sharing

VERIS is a framework for representing information security incidents based on the concepts of risk management. According to the SANS Institute (2015), VERIS is used by 20% of companies working with threat exchange technologies and therefore exhibits a significant practical relevance. Apart from describing actual incidents, it also covers the representation of estimations for possible impacts. Such additional information is intended to help determining the occurrence probabilities of specific incidents within certain companies. Therefore, an assessment of the risk level is supported too. VERIS is serialized by using a lightweight JSON format. Moreover, it provides various language-specific enumerations that enable unique definitions of components and fields such as types of malware.

Components: The base component of VERIS is the *incident* that describes the relation between *threat actors*, *actions* performed by those actors, information on *affected assets, attributes,* and an *impact assessment*. The **threat actor** is defined by a combination of its origin, motives and relation to the company. In this context, a differentiation between internal, external, and partner actors is introduced. **Actions** performed by those actors are specified by *tools* used for the attack combined with the actual *attack vector* and *vulnerabilities* exploited. **Affected assets** and **attributes** represent the counterpart to attacks by specifying information about the victimized objects as well as the protection goals that have been violated in this context. The **impact assessment** component provides a categorization of losses, loss estimations, and a rating of occurred impacts.

Besides the description of incidents, VERIS provides the additional intelligence components victim demographics, discovery, and response. Victim demographics provide further information about the origin of victims affected by an incident. This can, for example, define the geographical origin or an affiliation to a specific company or department. Discovery and response in turn provide the capability to express incidents, information about the discovery, identified root causes, attackers' intentions and defensive measures.

4.1.6. *X-ARF: Extended Abuse Reporting Format*

X-ARF is an approach for sharing threat information in a human readable form with the focus on performing the exchange process exclusively by using email. The practical relevance of this format can be derived from blocklist.de[4]. It is a free service that collects incident information in the X-ARF format and claims that the platform has close to 4000 reporting users till date. X-ARF is intended to be a lightweight and easy-to-use format that includes incident descriptions and built-in encryption capabilities. It uses email MIME extensions for the transport, YAML data structures to represent the information, and the JSON-schema to validate the contents.

Components: X-ARF defines very basic data structures that represent incidents as a combination of a basic incident description, incident category, information about the *attacker* as well as the *reporting entity*, the *affected system,* and the number of occurrences. These definitions are either defined as free-form text fields or text fields combined with basic enumerations defined within the format specification. Moreover, X-ARF messages support the Traffic Light Protocol[5], which can be used to ensure that information is shared with the appropriate audience. X-ARF also allows the inclusion of raw data, such as log files, to be transferred within incident reports (Kohlrausch et al., 2011).

4.1.7. *Further incident reporting formats*

In this section, we cover additional incident reporting formats having a certain relevance based on our point of view as well as the corresponding usage indicators that can be found in the literature. These formats are less relevant due to reasons like a minor practical usage or discontinued development and maintenance.

CISL (Common Intrusion Specification Language (Eckmann et al., 2002)) was defined as a format to exchange security-relevant information between intrusion detection systems. It was specified in 1998 within the Common Intrusion Detection Framework (CIDF) by a group of the University of Southern California. According to the authors, it is no longer an active area of work. However, some concepts have influenced other formats such as IODEF.

CAIF (Common Announcement Interchange Format (Goebel, 2005)) was defined as an extensible format for exchanging security announcements. It was specified in 2002 by a group called RUS-CERT of the University of Stuttgart. According to the authors, the development stopped in 2007.

ADeLe (Attack Description Language for Knowledge-Based Intrusion Detection (Cédric Michel, 2001)) was developed as a format for the structured exchange of threat information between intrusion detection systems. Alongside this exchange, the format was also designed to share system configurations to counter similar attacks. ADeLe is based on a concept from 2001, introduced by a group from the Supélec University in France, which is also the latest reference to this format that can be found in the literature. Therefore, it can be assumed that there is no further development in this area.

STATL (an Attack Language for State-based Intrusion Detection (Eckmann et al., 2002)) is also a format to exchange information between intrusion detection systems. However, the focus lies on the representation of specific system states as well as transitions that connect the system states. STATL was specified by a group from the University of Virginia in 2002.

ARF (Abuse Reporting Format[6]) was first introduced in 2005 by Yakov Shafranovich and then added to various IETF RFCs. This format allows the reporting of IT-security incidents in a structured manner and contributions were made in the recent past. However, since this is a highly specialized format for the reporting of email-related incidents, it will not be considered further within the scope of this work.

5. Criteria for evaluating incident reporting formats

The previous chapter introduced an overview of the relevant formats for the reporting of IT-security incidents. Within this chapter, we develop criteria for the comparison of these formats to build the basis for a later analysis. For this purpose, we propose criteria derived from the previously introduced UPSIDE pattern. Furthermore, we adapt criteria from the academic literature, adjusting them, wherever reasonable, to fit the specific purpose of incident reporting formats. Finally, we propose complementing criteria that are necessary for a comprehensive analysis from our point of view. This section concludes with a short overview of the developed criteria.

5.1. *Structural evaluation criteria*

While analyzing incident reporting formats, one basic aspect is the examination of the relationships between format's base structures and their components. This is especially important as these structural definitions specify all entities that can be represented. Therefore, they also define the contentual capabilities of each format and the use cases that can be covered. On these grounds, it is reasonable to derive criteria based on structural definitions and entity representations.

Previous work (Asgarli and Burger, 2016) showed that employing a quantitative analysis enables the qualification of statements about the **contentual coverage** of reporting formats. This includes basic information such as the extent, capacity, and granularity of the underlying data structures. From our point of view, this approach is reasonable for getting a general overview of the examined formats. Therefore, we adapt this criterion to our analysis. However, we think that qualitative aspects should also be considered for a comprehensive analysis.

Correspondingly, we propose to extend this quantitative approach by a qualitative format analysis by using the UPSIDE model. For criteria development, we first examine the core entities. Therefore, it is necessary to determine whether an explicit or implicit representation for the entity in question is present within a format. Implicit representations can, for example, be established by the combination of multiple entities. A further aspect in developing the criteria is to determine the granularity of the representations of each of the entities. Added together, they result in criteria that describe the core entities

[4] https://www.blocklist.de

[5] https://www.us-cert.gov/tlp

[6] https://tools.ietf.org/html/rfc5965

of the UPSIDE model regarding their presence and coverage. These include the entities **indicator**, **attacker**, **attack**, and **defender** at the top level as well as their respective child elements. The entity incident, however, will be excluded because it is already a necessary condition for reporting formats, as stated before.

From a methodological point of view, the sole presence of the basic reporting format elements, as described in UPSIDE, is a hard criterion with little room for interpretation. Therefore, it can directly be adapted as criterion for each element. In contrast, it is difficult to qualify general statements about the individual contentual coverage for each of the examined elements. Although UPSIDE defines the generic requirements for reporting formats and therefore allows an analysis and comparison of their core elements, a more detailed direct comparison of these components is questionable in many cases. This can be attributed to the deviating structures and approaches within the different reporting formats on lower level elements. Consequently, an individual contentual coverage will be defined for each of the elements described within UPSIDE. In addition, the quantitative approach introduced before gives an outline of the total structural depth and granularity of the formats.

5.2. *General evaluation criteria*

The previous section introduced the criteria to analyze incident reporting formats based on our generic incident model. Building on these criteria, we examine additional criteria for a later analysis.

Fenz et al. have analyzed the semantic potential of exchange standards and accordingly proposed several criteria for a comparison of these formats considering semantic aspects (Fenz et al., 2008). One of these aspects is the semantic usability that examines whether formats use structured formats like XML and thereby ensure the machine-readability as well as the presence of clear and unambiguous semantics. Although all the formats analyzed within this work are based on structured formats, it is still reasonable to adapt these criteria and distinguish between readability and semantics.

Machine-readability is one basic assumption that has to be made to enable automatic sharing of incidents. Although a format is based on a standardized language, it is necessary to inspect the structural extent of the applied format. This can be attributed to the fact that machine-readability can be considerably constrained by using, for example, free-text fields that allow the inclusion of unstructured information.

Human-readability is a criterion with a limited significance regarding automated threat sharing. However, since certain use cases require user interactions, this is also an important factor in case of a comprehensive analysis. This is especially the case if no tools for the evaluation of an incident reporting format are available, and therefore, a manual analysis is required.

Unambiguous semantics is another important criterion to be adapted to this analysis. Ambiguity within transferred data structures can have negative effects on machine-readability and interpretations of data.

Some further criteria for a comparison of security event exchange techniques have been suggested by Steinberger et al. (2015). These can also be adapted to the evaluation of incident reporting formats. These include the metrics interoperability, extensibility, aggregability, practical application, and human-readability, which will be shortly defined in the context of incident reporting formats.

Interoperability describes the capabilities of transferring data from one format into another without losing content or semantics. This is especially important since the sharing of threat information by definition requires multiple actors; these in turn might support deviating format standards. A low interoperability can therefore lead to problems while sharing incident information or even impede such sharing in various cases.

Extensibility describes the possibility of enriching the given exchange formats by additional data that has not been included in their base definitions. Additional data are an important factor as they enable the extension of formats with information about particular use cases that would otherwise have exceeded the format's capabilities. However, it must be considered that increased extensibility is always accompanied by drawbacks relating to machine-readability.

Aggregability describes the capabilities for the representation of multiple incidents and relations between them. It, therefore, indicates whether the exchange process could only cover isolated incident information or an overall view of the situation. Moreover, aggregating incident information generates additional information about incident coherences and thus can lead to improved incident handling in the long term.

Practical application refers to available tools that support an incident-sharing format and institutions that employ the respective format. Analogous to interoperability, practical application is a factor that indicates the operational capabilities of reporting formats. However, for some use cases, a high value in practical application increases the probability that the respective sharing partner would utilize the same format, which lowers the interoperability needs.

In addition to these criteria, Kampanakis (2014) have proposed the use of capabilities for the expression of external dependencies as a further criterion; we consider this reasonable and therefore adapt this criterion.

External dependencies are important as they create a common knowledge base for all participants in an incident-sharing process. This prevents any misunderstanding and ambiguities in the transferred data structures.

5.3. *Additional evaluation criteria*

In addition to the criteria based on the UPSIDE model as well as the introduced general criteria, we propose further criteria that are necessary for a complete analysis of reporting formats.

Licensing terms can have significant influence on the decision process for a format as well as on the usage of the format. This can be attributed to licensing terms that specify certain regulations for the utilization of the format. Such regulations may, for example, prohibit the customization of the format contents, thereby having a direct influence on the format's structural flexibility.

Maintenance efforts also constitute an important aspect in the analysis of reporting formats. Since the attacks and the victimized IT systems are evolving, it is necessary that representations for incidents evolve as well. Therefore, an

Table 2 – Evaluation criteria summary.

Structural evaluation criteria	General evaluation criteria	Additional evaluation criteria
Indicator	Machine-readability	Licensing terms
Attacker	Human-readability	Maintenance efforts
Attack	Unambiguous semantics	Documentation
Defender	Interoperability	
Contentual coverage	Extensibility	
	Aggregability	
	Practical application	
	External dependencies	

ongoing development of formats ensures the continuous capability of representing current threats. Maintenance efforts can be measured by the frequency of the released updates for the formats.

Documentation also has important implications for the implementation and usage of a format. Clearly documented structures ensure a syntactically and semantically correct communication between sharing partners and simplify implementations.

5.4. *Evaluation criteria summary*

Summarizing these findings, Table 2 illustrates an overview of the evaluation criteria introduced in this section. The columns of this table are arranged in accordance with the introduced criteria classification.

This overview is divided into the structural, general, and additional evaluation criteria. The column 'Structural evaluation criteria' shows the UPSIDE model base entities that have been supplemented by the contentual coverage as a further structural criterion. The column 'General evaluation criteria' gives an overview of those criteria that have been derived from the related literature. Conclusively, the column "Additional evaluation criteria" shows the supplementary criteria proposed within this paper. These criteria form the basis for the comparative analysis in the following section.

6. Comparative analysis of incident reporting formats

In the previous sections, we have introduced a generic model for incident reporting and important incident reporting formats. Moreover, we developed a set of structural criteria to compare incident reporting formats based on the UPSIDE model and criteria from the literature as well as the additionally proposed criteria. In this section, we apply the structural criteria, followed by the general and additional criteria, to each of the identified formats.

6.1. *STIX*

6.1.1. *Structural analysis*

STIX provides the representations of all components described in UPSIDE. **Incident** and **indicator** objects cover the root elements for an incident. The **attacker** component is described through the threat actor object, which can be attributed to single incidents or attack campaigns. It covers basic attributes and information about the attacker's current identity and objectives. **Objectives** are expressed as a combination of the underlying motivation and the intended effect supported by a measure of estimating the attacker's sophistication level. Although STIX does not express any direct representation of an **attack** object, representations for all defined attack components are provided. **Events** are expressed by observable objects by using the integrated CybOX language, which contains the representations of **actions taken** and **target objects**. **Methods and tools** are covered by TTP objects that describe attack patterns, malwares or exploits used within an attack. TTP objects also cover additional information about targeted victims and affected assets. Additionally, the exploit target object provides a representation of the assets exploited by these methods including the specifications for used **vulnerabilities** and weaknesses. An **attack result** can be defined within an incident object including a summary of the direct and the indirect impact as well as loss estimations. STIX also provides comprehensive representations of the **defender** reactions. These are represented by course of action objects, which define the actions for preventing or mitigating damage and information about the impact, cost, and efficiency of the performed actions.

Finally, STIX allows the representations for 286 object properties and datatypes (Asgarli and Burger, 2016); it, therefore, provides, especially compared to other format approaches, a comprehensive **contentual coverage** of incident information.

6.1.2. *General analysis*

The comprehensive contentual coverage and sparse use of free-text properties within STIX result in high **machine-readability**. However, these also result in weaknesses concerning the **human-readability** due to the inherent complexity. STIX exhibits clear structures and distinct object representations for incidents, indicators, and associated object entities, as described above. Therefore, it shows, depending on the use case, a very low potential for **ambiguous representations**. Conditioned by the comparatively large number of object properties and representations, STIX basically allows the expression of incidents and findings described in other formats. However, owing to format peculiarities, this is usually associated with the loss of information. The STIX repository also provides various tools for converting other formats into STIX. Altogether, this leads to a good **interoperability** valuation. Since STIX focuses on the automated sharing of threat information, it provides many invariant object properties. For these, no extension capabilities are intended due to a possible loss of compatibility, which in turn leads to low **extensibility**. Alongside the incident information, STIX also provides definitions of entire reports which can include multiple incidents, thereby inherently providing a good **aggregability** for incident information. Moreover, STIX is the de facto standard in incident sharing, as shown above, and is used in over 50 software products.[7] Therefore, it has the highest **practical application** of the examined formats. STIX also provides comprehensive support for the integration of **external dependencies** such as enumerations or incidents within remote systems or formats.

In addition, STIX provides a **permissive license** without copyleft restrictions. A precise and extensive **documentation** is available on the project's website. Moreover, the format repository is still actively **maintained** by OASIS. However, these efforts decreased with the introduction of STIX2.

6.1.3. *Summary*

STIX contains object representations for all UPSIDE components, including various additional elements, and therefore a comprehensive expression for each of the components. Only the references between elements and the representation for attacks do not fully match the model due to some differences within the base structure. STIX also fulfills most of the general criteria. It only shows weaknesses in areas of human-readability and extensibility due to structural guidelines defined within the language. Moreover, maintenance efforts have evidently decreased with the introduction of STIX2.

6.2. *STIX version 2*

6.2.1. *Structural analysis*

STIX2 provides data representations analogous to its predecessor and therefore structurally matches the UPSIDE model. STIX2 additionally provides an intrusion set element, which represents the attacker's behaviors and procedures. It matches the UPSIDE **attack** component, which cannot be directly represented by STIX. This is especially possible due to the introduced dynamic references of STIX2, which allows the expression of incidents aligned to the UPSIDE structure. Moreover, the implemented structural changes also result in additional top-level elements such as **attack pattern**, **tools**, and **vulnerability**, all of which are more accurate representations of UPSIDE components methods and tools as well as of vulnerability.

STIX2 provides a dynamic referencing structure that complicates an exact quantitative evaluation in terms of **contentual coverage**. However, owing to the use of the STIX base objects in combination with the dynamic referencing, it can be stated that STIX2 supports at least as much representation cases as STIX.

6.2.2. *General analysis*

The high number of possible representations in STIX2 leads to high **machine-readability** and low **human-readability** of the format. The use of dynamic references also enables the alignment of the format to specific use cases and therefore minimizes the potential of **ambiguities**. Similarly, this also enables the alignment of the format to the structure of other formats, thereby additionally increasing its **interoperability**. Analogous to STIX, there are no extensions for object entities intended. However, the use of dynamic references enables the coverage of additional use cases; this leads to a higher valuation of the **extensibility**. The **aggregation** of incidents is likewise supported. Owing to the recent release of STIX2 as well as the fact that only release candidates are currently available, it is unlikely that there is a **practical application** of the format at present. However, based on the wide propagation of STIX as well as the improvements made in the format structure, it can be assumed that STIX2 will soon gain significant practical importance. Analogous to its predecessor, STIX2 also supports **external dependencies**.

Furthermore, STIX2 provides a **permissive license**, high **maintenance** efforts, and a comprehensive **documentation**, all of which are identical to its predecessor STIX.

6.2.3. *Summary*

Introducing additional components and dynamic referencing, STIX2 allows the generation of incident representations that cover all UPSIDE components including the dependencies between them. STIX2 also fulfills most of the stated general criteria. Exceptions are human-readability and practical application. However, it is likely that the practical application will increase in the near future. In contrast to STIX, the dynamic structure of STIX2 provides a higher interoperability and extensibility.

6.3. *IODEF*

6.3.1. *Structural analysis*

IODEF does not support the representations of attacker or defender information, as shown in UPSIDE. Moreover, there is no utilization of indicator objects intended. However, IODEF provides multiple objects for the representation of **attacks,** which are directly integrated into the base object **incident**. This includes the representations of **event** actions and a description of the **affected target** by using system objects that allow a detailed specification of network nodes, systems, or services. IODEF provides an object method, which allows the expression of **methods and tools** used for an attack and basic information about the used **vulnerability**. Since the method object is specified as a free-text form element, the automatic processing of such information is aggravated. Moreover, IODEF utilizes an assessment object for representing an attack's **result** by expressing structured impact estimations based on various factors such as monetary and time factors.

IODEF provides the representations for 99 object properties and datatypes (Asgarli and Burger, 2016), and thereby offering a significantly lower **contentual coverage** compared to the STIX formats.

6.3.2. *General analysis*

IODEF extensively uses free-text representations for expressing additional incident information. This significantly lowers **machine-readability** capabilities, while **human-readability** is increased. IODEF provides overlapping elements such as incident and event data with similar representations (Fenz et al., 2008). Thus, it exhibits **ambiguity** problems concerning the format semantics. The extensive use of free-text fields leads

[7] https://wiki.oasis-open.org/cti/Products

to a high **interoperability** and **extensibility** of the format, since data structures that are not defined within IODEF can also be expressed. However, this leads to further losses in the structural expressiveness and automation capabilities. The format also supports the encapsulation of multiple incidents into one report and therefore provides a high **aggregability** analogous to STIX. As shown above, IODEF exhibits a high **practical application** with a share of 23% within sharing CERTs. One important drawback of this format is the missing support for **external dependencies**. Although it is theoretically possible to refer to enumerations within free-text fields, the approach allows no structured reference definitions.

IODEF is published under the IETF Trust Legal Provisions **(TLP) license**, which is an open source license based on the BSD license and hence offers unrestricted usage. Moreover, according to the format repository site[8], there were intense **maintenance** efforts, which have stopped with the introduction of IODEF2. The IODEF standard is entirely specified as IETF RFC. However, only little further **documentation** is available.

6.3.3. *Summary*

IODEF allows the expression of attack information within an incident, providing structured information about events and their results as well as additional unstructured information about the attack methods and vulnerabilities. However, indicators and perspectives for attackers and defenders are not covered. Moreover, IODEF moderately fulfills the stated general criteria. It shows the strengths for the criteria interoperability, extensibility, and aggregability especially due to the use of free-text fields, while it reveals the weaknesses concerning the criteria maintenance efforts and the integration of external dependencies.

6.4. *IODEF version 2*

6.4.1. *Structural analysis*

IODEF2 maintains the base structure and components of IODEF, which is extended by several additional objects and references. These include the representations of the **attacker** and **defender** perspectives in addition to the **attack** representations in IODEF. **Attacker** information can be expressed by using threat actor objects that are defined by free-text descriptions and can be consolidated into attack campaigns. However, attackers' objectives, as defined in the UPSIDE model, are not covered. By introducing the course of action properties, the expression of the **defender** perspective is also supported. However, in a free-text representation, IODEF2 introduces indicators that express detailed information about occurrences as well as possible attack phases and confidence levels. Moreover, the specifications of an attack result are extended by more detailed incident impact descriptions such as monetary or business impact.

IODEF2 provides similar object representations compared to its predecessor that were extended to a more holistic approach, as described before. IODEF2 provides 250 object properties and datatypes; this shows a huge increase in **contentual coverage** compared to the first version. This increase can be partly explained by the introduction of very specific event entities such as representations for digital signatures.

6.4.2. *General analysis*

IODEF2 clearly provides more extensive object representations that are still combined with free-text properties. Hence, it offers better **machine-readability** along with drawbacks in **human-readability** due to its increased complexity. This version also offers various changes to prevent **ambiguous** representations that were described for IODEF. Moreover, IODEF2 introduces the use of **external references**. Analogous to STIX2, it has also been published recently indicating a low **practical application**. Since this version is essentially an extension to its predecessor IODEF, the criteria **interoperability**, **extensibility**, and **aggregability** are to be evaluated analogously.

IODEF2 provides a **TLP license** and only little **documentation** are analogous to IODEF. However, owing to the replacement of IODEF, the IETF **maintenance** efforts changed over to IODEF2.

6.4.3. *Summary*

IODEF2 extends IODEF by additionally addressing the attacker and defender sides. Furthermore, it introduces the representations of indicators. It, therefore, nearly matches the UPSIDE model definitions except for expressing attackers' objectives. In contrast to IODEF, IODEF2 fairly fulfills most of the stated general criteria. It only shows the weaknesses in the criteria human-readability, documentation, and practical application.

6.5. *VERIS*

6.5.1. *Structural analysis*

VERIS provides the representations of **attackers** by using actor objects including specific attributes, suspected motives, and the attacker's origin. VERIS also supports the expression of **attack** information by using action objects, which distinguish between various types of attacks such as malware or hacking. The types of attacks in turn express free-text information about the attack, attack vectors, and the utilized **vulnerabilities**. VERIS also defines an attack result by using information about the compromise of assets and information about the estimated impact. The types of attacks provide an expression for the attack component, as described in UPSIDE. However, information other than the attack result is characterized by a few details. VERIS also covers information about the **defender** and its reactions. These include the discovery, evaluation, causes, and corrective actions. VERIS does not support the expression of indicator information within the base schema. However, the referencing of external **indicators** is possible for reasons of interoperability.

VERIS provides the representations of 137 object properties and datatypes that are divided into partly free-text properties and partly internal enumerations defined by the format. This leads to a structure that exhibits an average **contentual coverage** compared to the formats described before.

6.5.2. *General analysis*

As a result, the criteria **machine-readability** and **human-readability** can also be evaluated as average. VERIS provides

[8] https://datatracker.ietf.org/wg/mile/documents/

concise and clear information about attacks and the attacker. However, it provides comprehensive definitions for impact information, which can lead to **ambiguities** to a certain degree. It, for example, defines an overall impact amount as well as an impact loss amount, which inherently offers potential of ambiguity. Moreover, based on the application of free-text properties, VERIS partly offers a good **interoperability** and **extensibility** similar to the IODEF formats. However, there are some drawbacks due to sparsely defined attacks and very specific risk management properties. The **aggregability** of multiple incidents is not intended. With a propagation of 20%, VERIS offers a wide **practical application** similar to IODEF. Although the format definitions do not support **external dependencies**, the specification contains multiple internal enumerations for distinct definitions of incident information. However, these enumerations only represent generic information such as the malware definition "Worm", but they do not contain further information. Similar to IODEF, further information can be expressed by free-text properties, albeit without a pre-defined structure.

VERIS is published under the Creative Commons **ShareAlike (CC BY-SA) 4.0 license**. In contrast to the above-presented formats, it contains a strong copyleft that restricts possible use cases for the format. Referring to the format's repository, a continuous **maintenance** can be abstracted, although updates are performed in cycles of several months. Furthermore, VERIS offers a format specification included in its repository as well as a dedicated and comprehensive **documentation** on the format website.

6.5.3. *Summary*

VERIS allows the expression of all elements within the UPSIDE model with a focus on the representation of information about the attacker and the caused impact. It provides only superficial information about attacks. VERIS only moderately fulfills the stated general criteria. In particular, it shows special weaknesses in the areas of aggregability and external dependencies. However, it is worth highlighting that VERIS provides a comprehensive documentation.

6.6. *X-ARF*

6.6.1. *Structural analysis*

X-ARF is an approach that focuses on a simple implementation and therefore provides very basic capabilities for the representation of structured data. It allows the representation of basic **attack** information as well as information about the **attacker**. Indicator elements and information about defensive measures are not supported. Attacks are represented by a free-text description of the attack in combination with a description of the victimized system. **Methods and tools** are represented by a pre-defined X-ARF term such as 'fraud'. Using the traffic light protocol[9], the severity of an incident's result can be indicated. The attacker is expressed by identified information about the attack source such as the used IP-address. X-ARF also provides capabilities for appending information sources, such as log files, for a manual evaluation of incidents.

X-ARF represents the least extensive format with the lowest **contentual coverage** of 29 object properties and datatypes mostly expressed as free-text properties.

6.6.2. *General analysis*

X-ARF only provides very limited capabilities for an automated exchange of security information and low **machine-readability**, which is attributable to its minor complexity. However, this also leads to very good **human-readability**. Furthermore, the low level of complexity also leads to very low potential for **ambiguous** representations. Although basic information from other formats could be expressed, the **interoperability** can be valuated as very low as most information about extensive formats have to be expressed as free-text. These free-text fields also allow an **extension** of the format. However, extending basic information is only reasonable assuming a human receiver. Moreover, the format is intended to report single incidents; this implies that the **aggregation** of multiple incidents is not expressible. X-ARF also does not support **external dependencies**. It has a very high practical application with 3790 reporting users and 140,000 reported attacks on a daily basis.

X-ARF is published under a **license** without any restrictions. Despite the wide propagation, the format is only **maintained** on a yearly basis, and superficial **documentation** is provided.

6.6.3. *Summary*

X-ARF provides only basic elements for the representation of attacks and the attacker in combination with source file attachments. It is, therefore, not suited to express detailed incident information. X-ARF only poorly fulfills most of the stated general criteria. It only shows strengths in areas of human readability, ambiguous semantics, practical application, and licensing terms. However, owing to its extent and human-readability, it is well suited for a non-automated exchange.

6.7. *Summary of the analysis*

Summarizing the findings of this section, Table 3 illustrates an overview of the analysis results. It shows that the evaluated criteria as rows and the examined formats as columns. The analysis values are expressed as pie charts divided into four quarters, thus indicating the degree of fulfillment for each criterion.

This overview emphasizes that both STIX and STIX2 provide a comprehensive data model, low potential for ambiguity, a few extension capabilities, and high machine-readability. They are, therefore, best suited for an automated exchange of incident information. However, owing to the high complexity and low human readability, a manual interpretation is only reasonably supported by a software-based approach. They also provide a very high practicability due to the high interoperability, unrestrictive license terms, continuous maintenance and practical application of STIX, and presumably, STIX2 in future. Moreover, subsequent threat analysis efforts are supported by the utilization of external dependencies and the aggregability of threat information.

IODEF in turn explicitly enables the exchange of attack information in a semi-structured manner. This leads to an

[9] https://www.us-cert.gov/tlp

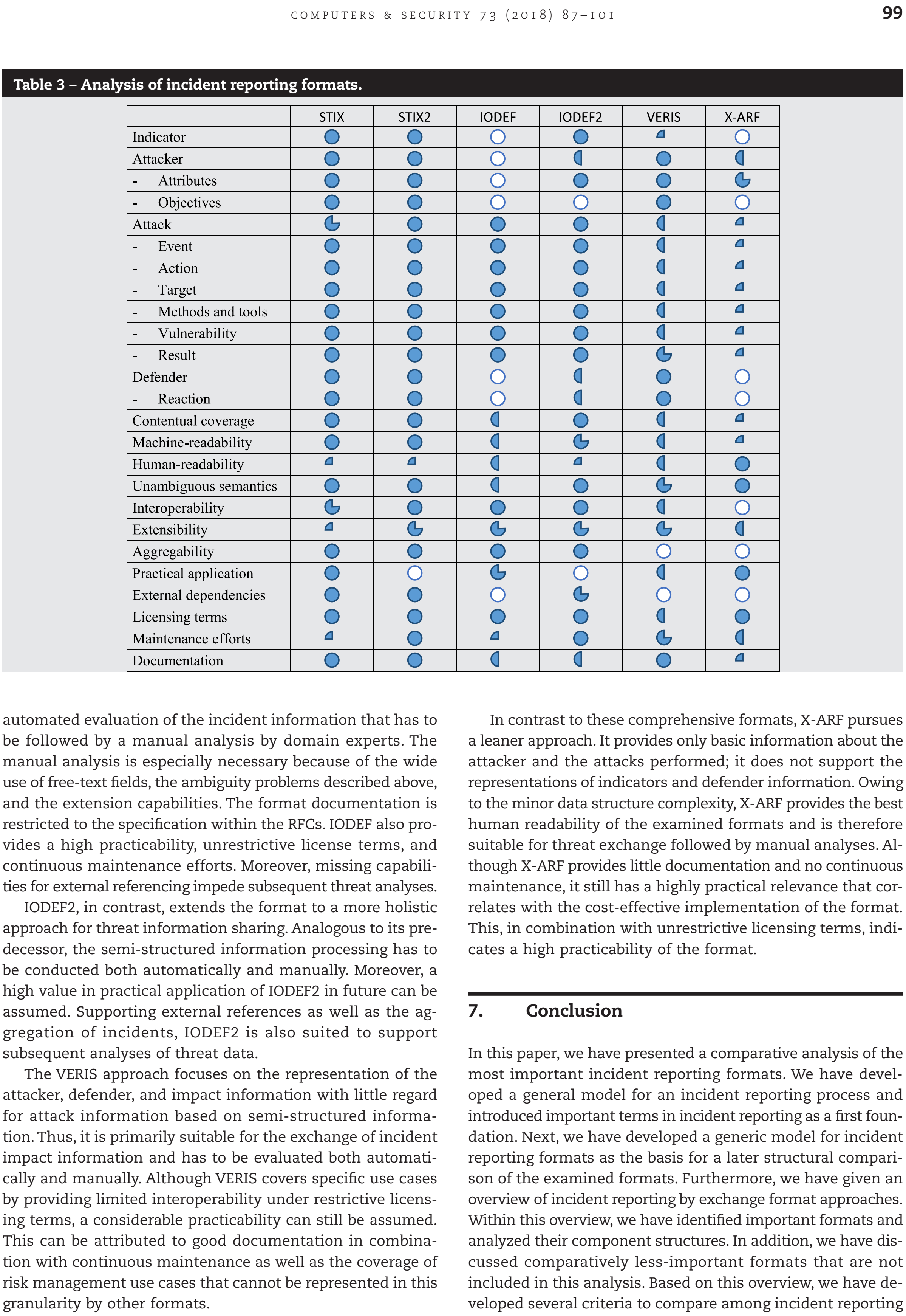


Table 3 – Analysis of incident reporting formats.

	STIX	STIX2	IODEF	IODEF2	VERIS	X-ARF
Indicator	●	●	○	●	◔	○
Attacker	●	●	○	◑	●	◑
- Attributes	●	●	○	●	●	◕
- Objectives	●	●	○	○	●	○
Attack	◕	●	●	●	◑	◔
- Event	●	●	●	●	◑	◔
- Action	●	●	●	●	◑	◔
- Target	●	●	●	●	◑	◔
- Methods and tools	●	●	●	●	◑	◔
- Vulnerability	●	●	●	●	◑	◔
- Result	●	●	●	●	◕	◔
Defender	●	●	○	◑	●	○
- Reaction	●	●	○	◑	●	○
Contentual coverage	●	●	◑	●	◑	◔
Machine-readability	●	●	◑	◕	◑	◔
Human-readability	◔	◔	◑	◔	◑	●
Unambiguous semantics	●	●	◑	●	◕	●
Interoperability	◕	●	●	●	◑	○
Extensibility	◔	◕	◕	◕	◕	◑
Aggregability	●	●	●	●	○	○
Practical application	●	○	◕	○	◑	●
External dependencies	●	●	○	◕	○	○
Licensing terms	●	●	●	●	◑	●
Maintenance efforts	◔	●	◔	●	◕	◑
Documentation	●	●	◑	◑	●	◔

automated evaluation of the incident information that has to be followed by a manual analysis by domain experts. The manual analysis is especially necessary because of the wide use of free-text fields, the ambiguity problems described above, and the extension capabilities. The format documentation is restricted to the specification within the RFCs. IODEF also provides a high practicability, unrestrictive license terms, and continuous maintenance efforts. Moreover, missing capabilities for external referencing impede subsequent threat analyses.

IODEF2, in contrast, extends the format to a more holistic approach for threat information sharing. Analogous to its predecessor, the semi-structured information processing has to be conducted both automatically and manually. Moreover, a high value in practical application of IODEF2 in future can be assumed. Supporting external references as well as the aggregation of incidents, IODEF2 is also suited to support subsequent analyses of threat data.

The VERIS approach focuses on the representation of the attacker, defender, and impact information with little regard for attack information based on semi-structured information. Thus, it is primarily suitable for the exchange of incident impact information and has to be evaluated both automatically and manually. Although VERIS covers specific use cases by providing limited interoperability under restrictive licensing terms, a considerable practicability can still be assumed. This can be attributed to good documentation in combination with continuous maintenance as well as the coverage of risk management use cases that cannot be represented in this granularity by other formats.

In contrast to these comprehensive formats, X-ARF pursues a leaner approach. It provides only basic information about the attacker and the attacks performed; it does not support the representations of indicators and defender information. Owing to the minor data structure complexity, X-ARF provides the best human readability of the examined formats and is therefore suitable for threat exchange followed by manual analyses. Although X-ARF provides little documentation and no continuous maintenance, it still has a highly practical relevance that correlates with the cost-effective implementation of the format. This, in combination with unrestrictive licensing terms, indicates a high practicability of the format.

7. Conclusion

In this paper, we have presented a comparative analysis of the most important incident reporting formats. We have developed a general model for an incident reporting process and introduced important terms in incident reporting as a first foundation. Next, we have developed a generic model for incident reporting formats as the basis for a later structural comparison of the examined formats. Furthermore, we have given an overview of incident reporting by exchange format approaches. Within this overview, we have identified important formats and analyzed their component structures. In addition, we have discussed comparatively less-important formats that are not included in this analysis. Based on this overview, we have developed several criteria to compare among incident reporting formats by considering both explicitly structural and general aspects. Using these criteria, we have also conducted a comprehensive analysis for each of the examined formats, followed by a final overview and format classification.

The analysis reveals strengths, weaknesses, and additional information for the comparison of the formats. It also explains that each format has a use case that it fits in the best possible manner and therefore has a right to exist. Based on these findings, our future work will focus on the research on approaches and methodologies for the automated processing of incident reporting formats as well as the application of exchanged data.

Acknowledgment

This work is performed under the BMBF-DINGfest project which is supported under contract by the German Federal Ministry of Education and Research (16KIS0501K). The DINGfest project is dedicated to the detection of malicious system states in combination with visualization, forensic preparation and reporting of identified IT-security incidents. As part of the project, this work builds the basis for the definition of incident information sharing requirements as well as for the decision of a suitable incident reporting format.

REFERENCES

Asgarli E, Burger E. Semantic ontologies for cyber threat sharing standards. IEEE Sympo Technol Homeland Secur (HST) 2016;doi:10.1109/THS.2016.7568896.

Barnum S. Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX), 2014. Available from: http://stixproject.github.io/getting-started/whitepaper/.

Blackwell C. A Security Ontology for Incident Analysis, In: CSIIRW '10 Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research, 2010, doi:10.1145/1852666.1852717.

Cédric Michel LM. Adele: An Attack Description Language For Knowledge-Based Intrusion Detection, In: IFIP/Sec '01 Proceedings of the IFIP TC11 Sixteenth Annual Working Conference on Information Security – Trusted Information: The New Decade Challenge, pp. 353–368, 2001, doi:10.1007/0-306-46998-7_25.

Ciardhuáin S. An extended model of cybercrime investigations. Int J Digit Evid 2004;3:1–22. doi:10.1504/IJESDF.2010.033780. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.80.1289&amp;rep=rep1&amp;type=pdf%5Cnhttps://utica.edu/academic/institutes/ecii/publications/articles/A0B70121-FD6C-3DBA-0EA5C3E93CC575FA.pdf.

Cichonski P, Millar T, Grance T, Scarfone K. Computer Security Incident Handling Guide: Recommendations of the National Institute of Standards and Technology, In: NIST Special Publication Volume: 800-61 Revision 2, pp. 79, 2012, doi:10.6028/NIST.SP.800-61r2.

Congress of the United States of America, National Cybersecurity and Critical Infrastructure Protection Act of 2014, 2014. Available from: https://www.congress.gov/bill/113th-congress/house-bill/3696/text.

Dandurand L, Kaplan A, Kácha P, Kadobayashi Y, Kompanek A, Lima T. Standards and tools for exchange and processing of actionable information, pp. 51, ISBN: 9789292041052, 2014.

Deniz O, Celikoglu HB. Overview to some existing incident detection algorithms: a comparative evaluation. Procedia Soc Behav Sci 2011;0:1–13.

Deutscher Bundestag, Gesetz zur Erhöhung der Sicherheit informationstechnischer Systeme, 2015. Available from: https://www.bmi.bund.de/SharedDocs/Downloads/DE/Gesetzestexte/it-sicherheitsgesetz.pdf.

Eckmann S, Vigna G, Kemmerer R. STATL: an attack language for state-based intrusion detection. J Comput Secur 2002;10(1–2):71–104. doi:10.3233/JCS-2002-101-204. Available from: http://iospress.metapress.com/index/FE5HUFTEYU5BE7VW.pdf.

ENISA, Detect, SHARE, Protect Solutions for Improving Threat Data Exchange among CERTs. In: pp. 51, 2013.

ENISA, Good practice guide for incident management. pp. 60, 2010. Available from: https://www.enisa.europa.eu/publications/good-practice-guide-for-incident-management/at_download/fullReport.

European Commission, NIS Directive 2016/1148 (EU) of the European Parliament and of the Council, 2016. Available from: http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016L1148.

Fenz S, Ekelhart A, Weippl E. Semantic Potential of existing Security Advisory Standards, In: Inproceedings of the FIRST 2008, pp. 1–8, 2008. Available from: https://tinyurl.com/mg63q33.

Freiling FC, Schwittay B. A Common Process Model for Incident Response and Computer Forensics. Imf 2007;19–40. Available from: http://www1.cs.fau.de/filepool/publications/imf2007-common-model.pdf.

Goebel O. Common announcement interchange format (CAIF) version 1.2, 2005. Available from: http://www.caif.info/draft-goebel-caif-format.html.

Howard JD, Longstaff TA. A common language for computer security incidents, 1998, doi:10.2172/751004. Available from: http://prod.sandia.gov/techlib/access-control.cgi/1998/988667.pdf.

IBM Corporation, Indicators of compromise, 2015. Available from: https://pcatt.org/techblog/wp-content/uploads/2015/10/IndicatorsOfCompromise.pdf.

Johnson C. Failure in Safety-Critical Systems: A Handbook of Incident and Accident Reporting, pp. 86–90, ISBN: 0852617844, 2003.

Kampanakis P. Security automation and threat information-sharing options. IEEE Secur Priv 2014;12(5):42–51. doi:10.1109/MSP.2014.99.

Kohlrausch J, Übelacker S, G. Jra, T. Internal, X-ARF: A Reporting and Exchange Format for the Data Exchange of Netflow and Honeypot Data, 2011. Available from: http://geant3.archive.geant.net/Media_Centre/Media_Library/MediaLibrary/xarf_geant_milestone2.pdf.

McAffee Corporation, McAfee Labs – Threat-Report, 2016. Available from: https://www.mcafee.com/de/resources/reports/rp-quarterly-threats-mar-2016.pdf.

MITRE Corporation, Making Security Measurable, In: Proceedings – IEEE Military Communications Conference MILCOM, pp. 1–9, ISBN: 9781424426775, 2011.

Munteanu VI, Edmonds A, Bohnert TM. Cloud Incident Management. Challenges, Research Directions and Architectural Approach, In: Proceeding UCC '14 Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing Pages 786–791, 2014, pp. 0–5, 2014, http://ieeexplore.ieee.org/document/7027595/

Pirc J. Threat Forecasting: Leveraging Big Data for Predictive Analysis, pp. 48–54, ISBN: 978128000069, 2016.https://www.amazon.de/Threat-Forecasting-Leveraging-Predictive-Analysis/dp/012800006

Prosise C, Kevin M. Incident Response & Computer Forensics, Second Edition, pp. 548, 2003, doi:10.1036/0072230371.

Sauerwein C, Sillaber C, Mussmann A, Breu R. Threat Intelligence Sharing Platforms: An Exploratory Study of Software Vendors and Research Perspectives, In: 13. Internationale Tagung Wirtschaftsinformatik, WI 2017, St. Gallen, 2017.

SANS Institute, Who's Using Cyberthreat Intelligence and How?, 2015. Available from: https://www.sans.org/reading-room/whitepapers/analyst/cyberthreat-intelligence-how-35767.

Steinberger J, Sperotto A, Golling M, Baier H. How to exchange security events? Overview and evaluation of formats and protocols, In: IEEE International Symposium on Integrated Network Management (IM), pp. 261–269, ISBN: 9783901882760, 2015.

Symantec Corporation, Internet Security Threat Report, 2016. Available from: http://linkinghub.elsevier.com/retrieve/pii/S1353485805001947.

Wunder J, Struse R, Jordan B, Piazza R. STIX^{TM} Version 2.0, OASIS CTI, 2017. Available from: https://docs.google.com/document/d/1IcA5KhglNdyX3tO17bBluC5nqSf70M5qgK9nuAoYJgw.

Yusoff Y, Ismail R, Hassan Z. Common phases of computer forensics investigation Models. Int J Computer Sci Inf Technol 2011;3:17–31. doi:10.5121/ijcsit.2011.3302.

Günther Pernul received both the diploma degree and the doctorate degree (with honors) from the University of Vienna, Austria. Currently he is full professor at the Department of Information Systems at the University of Regensburg, Germany. Prior he held positions with the University of Duisburg-Essen, Germany and with University of Vienna, Austria, and visiting positions the University of Florida and the College of Computing at the Georgia Institute of Technology, Atlanta. His research interests are manifold, covering data and information security aspects, data protection and privacy, data analytics, and advanced data centric applications.

Florian Menges received both the Bachelor of Science and Master of Science degree from the University of Regensburg, Germany. Currently he is research assistant at the Department of Information Systems at the University of Regensburg, Germany. His research interests include threat intelligence with a focus on sharing and reporting intelligence data, storage strategies for intelligence data as well as anonymization techniques and incentivizing the sharing and reporting of incident data.

2 Unifying Cyber Threat Intelligence

Current status:	Published	
Conference:	Trust, Privacy and Security in Digital Business - 16th International Conference, TrustBus, Linz, Austria, August 26-29, 2019	
Date of acceptance:	31 May 2019	
Full citation:	Florian Menges, Christine Sperl and Günther Pernul. Unifying Cyber Threat Intelligence. *Gritzalis S., Weippl E., Katsikas S., Anderst-Kotsis G., Tjoa A., Khalil I. (eds) Trust, Privacy and Security in Digital Business. TrustBus 2019. Lecture Notes in Computer Science, vol 11711. Springer, Cham (2019).*	
Authors contributions:	Florian Menges	70%
	Christine Sperl	20%
	Günther Pernul	10%

Conference Description: TrustBus brings together researchers from different disciplines, developers, and users all interested in the critical success factors of digital business systems. TrustBus offers a platform for the scientific communication of papers, work-in-progress reports, and industrial experiences describing advances in all areas of digital business applications related to trust and privacy. TrustBus proceedings are published in the *Springer Lecture Notes in Computer Science*.

Unifying cyber threat intelligence

Florian Menges, Christine Sperl, and Günther Pernul

University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany

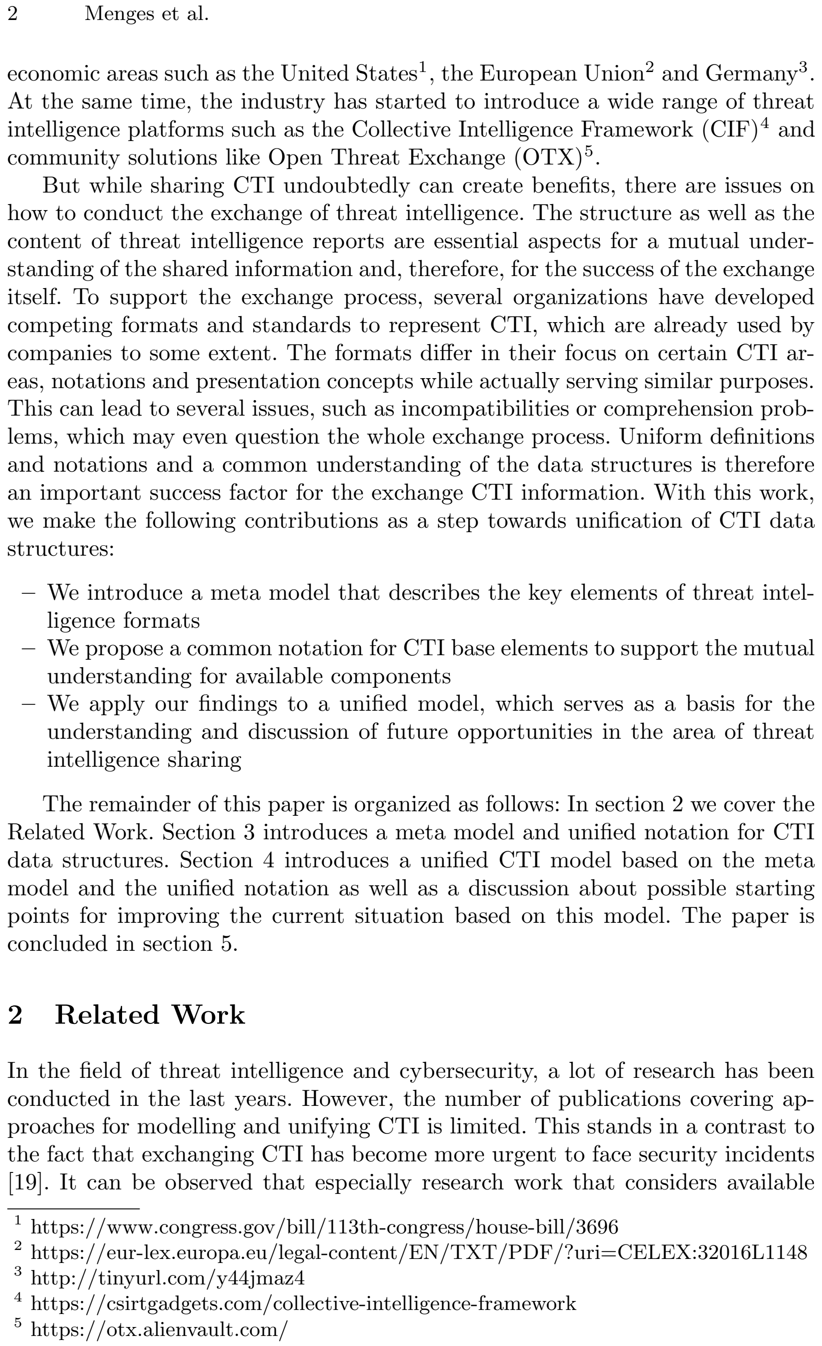
Abstract. The threat landscape and the associated number of IT security incidents are constantly increasing. In order to address this problem, a trend towards cooperative approaches and the exchange of information on security incidents has been developing over recent years. Today, several different data formats with varying properties are available that allow to structure and describe incidents as well as cyber threat intelligence (CTI) information. Observed differences in data formats implicate problems in regard to consistent understanding and compatibility. This ultimately builds a barrier for efficient information exchange. Moreover, a common definition for the components of CTI formats is missing. In order to improve this situation, this work presents an approach for the description and unification of these formats. Therefore, we propose a model that describes the elementary properties as well as a common notation for entities within CTI formats. In addition, we develop a unified model to show the results of our work, to improve the understanding of CTI data formats and to discuss possible future research directions.

Keywords: Incident reporting · Incident management · Incident response · Reporting formats · STIX · IODEF · VERIS.

1 Motivation

In the age of digitization, information systems play a more integrated and important role in modern society than ever before. This also applies to critical infrastructures that are essential for the functioning of society today. At the same time, however, these systems are becoming increasingly complex and vulnerable to attacks. It can be observed that today's systems are mostly defended by traditional security measures that only provide basic protection against common threats. In contrast, reliable protection of systems against sophisticated and targeted attacks remains a problem and continues to intensify the arms race between threat actors on the one side and security experts on the other. To be ahead in this game, a trend towards the exchange of Cyber Threat Intelligence (CTI) information has emerged to be aware of threats at an early stage. This can either strengthen threat prevention or contribute to the mitigation of already occurred incidents and improve the overall system security.

The benefits of CTI exchanges are recognized and promoted by various governments, industry and research. This has already led to legal reporting obligations for industries that are relevant to the functioning of the society in different



economic areas such as the United States[1], the European Union[2] and Germany[3]. At the same time, the industry has started to introduce a wide range of threat intelligence platforms such as the Collective Intelligence Framework (CIF)[4] and community solutions like Open Threat Exchange (OTX)[5].

But while sharing CTI undoubtedly can create benefits, there are issues on how to conduct the exchange of threat intelligence. The structure as well as the content of threat intelligence reports are essential aspects for a mutual understanding of the shared information and, therefore, for the success of the exchange itself. To support the exchange process, several organizations have developed competing formats and standards to represent CTI, which are already used by companies to some extent. The formats differ in their focus on certain CTI areas, notations and presentation concepts while actually serving similar purposes. This can lead to several issues, such as incompatibilities or comprehension problems, which may even question the whole exchange process. Uniform definitions and notations and a common understanding of the data structures is therefore an important success factor for the exchange CTI information. With this work, we make the following contributions as a step towards unification of CTI data structures:

- We introduce a meta model that describes the key elements of threat intelligence formats
- We propose a common notation for CTI base elements to support the mutual understanding for available components
- We apply our findings to a unified model, which serves as a basis for the understanding and discussion of future opportunities in the area of threat intelligence sharing

The remainder of this paper is organized as follows: In section 2 we cover the Related Work. Section 3 introduces a meta model and unified notation for CTI data structures. Section 4 introduces a unified CTI model based on the meta model and the unified notation as well as a discussion about possible starting points for improving the current situation based on this model. The paper is concluded in section 5.

2 Related Work

In the field of threat intelligence and cybersecurity, a lot of research has been conducted in the last years. However, the number of publications covering approaches for modelling and unifying CTI is limited. This stands in a contrast to the fact that exchanging CTI has become more urgent to face security incidents [19]. It can be observed that especially research work that considers available

[1] https://www.congress.gov/bill/113th-congress/house-bill/3696

[2] https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016L1148

[3] http://tinyurl.com/y44jmaz4

[4] https://csirtgadgets.com/collective-intelligence-framework

[5] https://otx.alienvault.com/

CTI data formats and the underlying data structures is rare. Most common in the area of modelling and unifying CTI are ontology proposals that clarify terms and their relations to each other in a defined area. Ontologies that can be found in literature can be distinguished in specialized and generic approaches.

Specialized approaches describe specific aspects or elements of threat intelligence as shown in the following. Falk and Way present an ontology that focuses on threat actors [6] and Grgio et al. propose an ontology that describes suspicious behavior of malware [9]. These works focus on particular threat intelligence aspects, while the big picture of CTI is not covered.

Generic approaches on the contrary have a broader scope. Fenz et al. perform valuable fundamental work prior to the establishment of most of the CTI sharing formats. They define syntax relations of different security concepts based on existing guidelines like the German IT Grundschutz Manual [7]. Falk proposes a threat intelligence ontology utilizing the Lockheed Martin Cyber Kill Chain in combination with events and threat actors. Although providing a broader scope, this work also does not fully cover the aspects of CTI or its data structures [5]. Iannacone et al. create a graph-based ontology for representing threat intelligence information, which provides a broader view on different aspects of threat intelligence. However, it shows a clear focus on specific attacks and lacks important CTI concepts such as attacker techniques and campaigns and does not consider underlying data formats other than STIX[6] [11]. Oltramari et al. focus on combining human and machine elements to create a cyber security ontology offering a detailed description of the included elements. Nevertheless, the research lacks a specific relation to CTI as well as the consideration of relevant data formats [18].

Other research work focuses on common specifications in the field of threat intelligence. First Howard and Longstaff establish a widely known and accepted language for computer security incidents where basic terms of an incident are defined [10]. Burger et al. create a taxonomy model for cyber threat intelligence sharing which organizes the different formats like IODEF[7] and STIX in the categories Transport, Session, Indicators, Intelligence and Attributes [4]. Mavroeidis and Bromander develop a model to compare the different taxonomies, ontologies and sharing formats to enable the finding of further research areas [14]. However, these approaches neither cover the basic properties of CTI formats nor their specific notation elements, as we provide in this paper.

A further research direction focuses on approaches that incorporate specific data formats for representing CTI information. Obrst et al. utilize different standards such as MAEC[8], CEE[9], Cybox[10] and STIX, to create a comprehensive cyber ontology. Although, this research gives a broad overview on CTI data structures, it is limited to formats related to STIX. Moreover, data types, at-

[6] https://stixproject.github.io/about/STIX_Whitepaper_v1.1.pdf

[7] https://tools.ietf.org/html/rfc5070

[8] https://maecproject.github.io/

[9] https://cee.mitre.org

[10] https://cyboxproject.github.io/

tributes and notation issues are not covered [17]. Zhao et al. propose a unified representation of CTI using an ontology based model for threat intelligence built on the study of security incidents and on elements of STIX2[11]. This model allows a more specific representation of threat intelligence data. However, as it only incorporates STIX2, it does not cover differences between CTI data formats [23].

Summarizing, it can be stated that the literature for CTI models is limited. Predominant are ontologies, which can be categorized in specialized and holistic. Other approaches focus on structuring and comparing different concepts. Some works cover data formats but a comprehensive view on CTI data structures from different sources is lacking. Moreover, attempts in finding a common notation and creating a unified models don not incorporate all the relevant incident sharing data formats. Even though all these approaches contribute their part to generate insight to threat intelligence information, a comprehensive view on CTI data structures from different sources has not been conducted yet. A meta model considering the syntax from different threat intelligence sharing formats and a holistic view with integrated data formats both are currently missing. Moreover, there is no academic work covering key elements of CTI sharing formats or unifying their notations to the best of our knowledge.

3 A standardized representation for Threat Intelligence Information

A successful exchange of cyber threat intelligence strongly depends on a mutual understanding of contents shared by the parties involved. The heterogeneous nature of sharing formats is one of the main barriers in sharing this data. Therefore, it is important to find an agreement on basic terms as well as unified definitions for shared elements. To build a foundation for such an agreement, we propose a meta model as a guidance for the modelling, classification and comparison of CTI formats in this section. We also propose a standardization of threat intelligence elements. This includes a unified nomenclature and classification for CTI elements as a step towards the homogenization of CTI data formats.

3.1 Meta model for threat intelligence information

As a first step we aim to create a common understanding of relevant concepts for the representation of CTI. Therefore, we introduce a meta model that provides a comprehensive specification, covering both the basic structuring elements and coherences that can be used to express intelligence information. The model is intended to support the verification and extension of existing models as well as the creation of further model instances. It also serves as a basis for understanding elements and relationships within existing formats.
From a methodical perspective, the developed model is following the archetypal abstraction concepts for meta modeling by Sprinkle et al. [20]. According to this

[11] https://oasis-open.github.io/cti-documentation/

concept, the elements Class, Association, Specialization and Constraint are used to compose the model. To realize a more accurate representation for CTI concepts, Association elements are further detailed into the elements Composition and Aggregation. This allows an additional differentiation between mandatory and optional relationships and therefore increases the models expressiveness.

The meta model (see figure 1) is developed based upon the characteristics of state-of-the-art formats for structured CTI representation STIX1/2, IODEF and IODEF2[12], VERIS[13] and X-ARF[14] as outlined in Menges and Pernul [16]. Further formats that can be found in literature, such as MISP[15], openIoC [16] are excluded from the development process. This is mainly due to their limited data model and focus on threat intelligence events and indicators as outlined by Burger et al. [4]. The model development process is realized in two consecutive steps. First, the relevant literature in the areas of CTI in general and state-of-the-art formats is reviewed. Within this review, important concepts for representing CTI are identified. In the second step, the structures and characteristics of these formats are analyzed to validate and supplement the insights gained from the literature review. This process results in the identification of fundamental concepts such as elements, properties and relationships of CTI. These concepts are then translated into appropriate meta-types that are finally combined to build the CTI meta model.

First, we discuss the different aspects important for representing CTI that are derived from the literature. These aspects are translated into the first meta model building blocks. All of the examined formats define a set of base entities, each of which represents one of its core components in an object oriented way as shown by Bourgue et al. [2]. The resulting Object entities enable the fundamental representation of CTI data and attributes. The formats also define capabilities to introduce Relationships between these objects. In addition, the examined formats define one distinct root element that collects all base entities into a reportable collection, which we define as Report in the proposed model. Burger et al. show that basic CTI objects can be assigned to the three different categories Indicator, Intelligence and Attribution [4]. For the development of our model, we translate these assignments into three kinds of classes that inherit from the base class Object that are defined in the following.

Indicator objects describe patterns or behaviors that show the likelihood that an incident is occurring, has already occurred or will probably occur in the future. This includes representations for genuine system observations as well as indication objects for structuring observations and assessing the probability of them being part of an incident.

Intelligence objects are used to represent specific knowledge about threats or incidents. This includes the combination of findings from indicators or past oc-

[12] https://tools.ietf.org/html/rfc7970

[13] http://veriscommunity.net

[14] http://xarf.org/

[15] https://www.misp-project.org/

[16] https://github.com/mandiant/OpenIOC_1.1

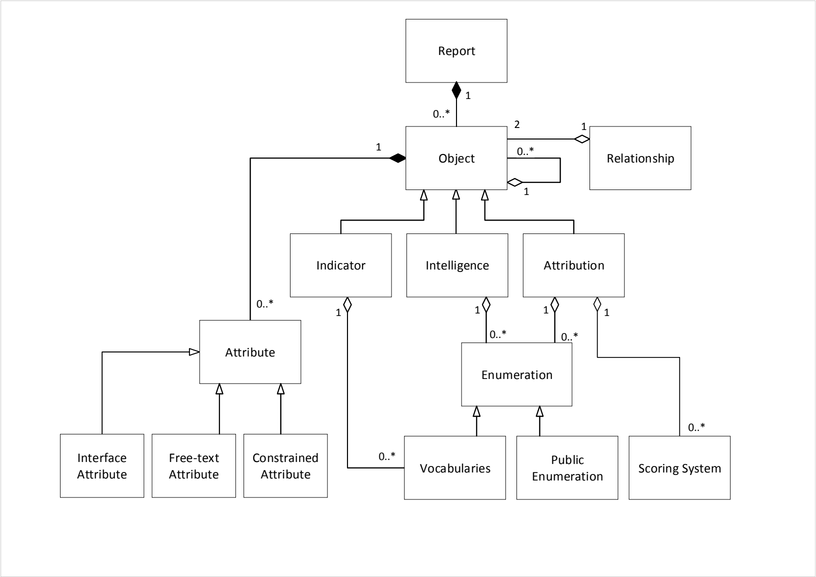


Fig. 1. CTI Meta model

currences to derive a specific threat behavior or attack pattern. Moreover, the intelligence objects also define specific actions and countermeasures for known threats that can be performed for the prevention of incidents or impact mitigation.

Attribution objects describe the source, the target as well as the circumstances of an incident. More precisely, the source defines both information about entities involved in an incident and their possible location, objectives or interests. The target includes entities affected by an incident including vulnerabilities that are used to compromise the entity. Moreover, Attribution also describes the circumstances of an incident, including information such as the effects of an incident or the start and end of an attack.

The described objects serve as a structural foundation for describing CTI information. To cover all aspects of representing CTI information, the defined object attribute definitions also have to be examined and differentiated in detail. As shown by Burger et al. [4], Steinberger et al. [21] as well as Menges and Pernul [16], the specification and usage of object attributes is decisive for the interoperability of CTI formats. More precisely, there are two main types of attribute definitions: Constrained Attributes and Free-text Attributes. Constrained Attributes usually lead to structured and comprehensible information due to their underlying guidelines. Free-text Attributes without guidelines, however, lead to unpredictable and non-structured information that reduces the format interoperability and automation capability. Besides these attributes, the model also includes an Interface Attribute, which is derived from the format specifications and, therefore, introduced with the specification analysis of this section.

Another vital aspect for CTI formats is the integration of interoperability frameworks to ensure a mutual understanding of the represented contents. Syed et al. [22] consider interoperability frameworks as integral elements within their proposed threat intelligence ontology. Mavroeidis and Jøsang [15] also underline the central role of these frameworks for structuring threat information. The frameworks include Vocabularies, Public Enumerations and Scoring Systems, which are tied to CTI objects. In the following, we provide a short description of these frameworks underlining their contribution for CTI interoperability and automation capabilities.

Vocabularies (also internal Enumerations) represent lists of predefined content for object attributes that are supplied with CTI formats. An example for a vocabulary is SecurityCompromiseVocab provided by STIX. It defines the possible values *Yes*, *Suspected*, *No* and *Unknown* as allowed attribute contents for the attribute SecurityCompromise within an incident object. It therefore provides a clear field definition based on predefined values, preventing ambiguities. This enables a common understanding for contents that are expressed using vocabularies. A more detailed introduction to the application of vocabularies within the exchange of CTI is given by Fransen et al. [8], while emphasizing their contribution to the automation capabilities of CTI.

Public Enumerations provide publicly available registers that can be used to clearly specify particular CTI aspects such as configurations, platforms, weaknesses or vulnerabilities. An exemplary enumeration is Common Vulnerabilities and Exposures (CVE)[17], which provides a broad collection of uniquely identifiable vulnerability definitions and descriptions for different systems. CVE is publicly administered and therefore available for any participant in an intelligence exchange process. Using this enumeration, vulnerabilities can be clearly described within a CTI format using a reference to its unique identifier provided by the enumeration. The importance of integrating enumerations into CTI formats and the accompanying benefits of interoperability and automation capabilities are described by Brown et al. [3].

Scoring Systems provide a consistent method to capture the characteristics of particular threat intelligence aspects, mapping them into quantitative descriptions. More precisely, numerical values are generated from the underlying information, enabling the assessment and comparability of the information. An exemplary system is the Common Vulnerability Scoring System (CVSS)[18]. It allows to capture and assess the characteristics of a vulnerability. CVSS provides different calculation rules that quantify the gathered vulnerability information and translates it into a numerical score reflecting the severity of the vulnerability. Integrated into threat intelligence standards, Scoring Systems provide a structured way to express ratings and assessments in a common understandable and

[17] https://cve.mitre.org/

[18] https://www.first.org/cvss/

interoperable form. The importance of these systems is also shown by Brown et al. [3] and Kampanakis [12] underlining their contribution to the interoperability of CTI. The literature based elements of the meta model, are now validated, supplemented and relationsships are established using the specifications of the considered CTI formats.

When comparing the defined objects with the format specification, the central role of the Report element is confirmed. It serves as the base component to reference the core CTI objects. Since the specifications basically allow to create empty reports, the relationship between Report and the core objects is defined as an aggregation. Although, all formats allow to build relationships between objects, there are particular differences between them. Comparing the formats, there are two different types of possible relationships. On the one hand, relationships can be expressed as single objects that provide attributes for defining the objects to be connected, which is for example the case for STIX2. Their relationship is defined as a composition, since relationship objects cannot exist without their referenced entities. On the other hand, references can also be defined using attributes within the objects to be connected, which is for example the case for IODEF. This relationship on the contrary is defined as aggregation, due to the optional nature of their references.

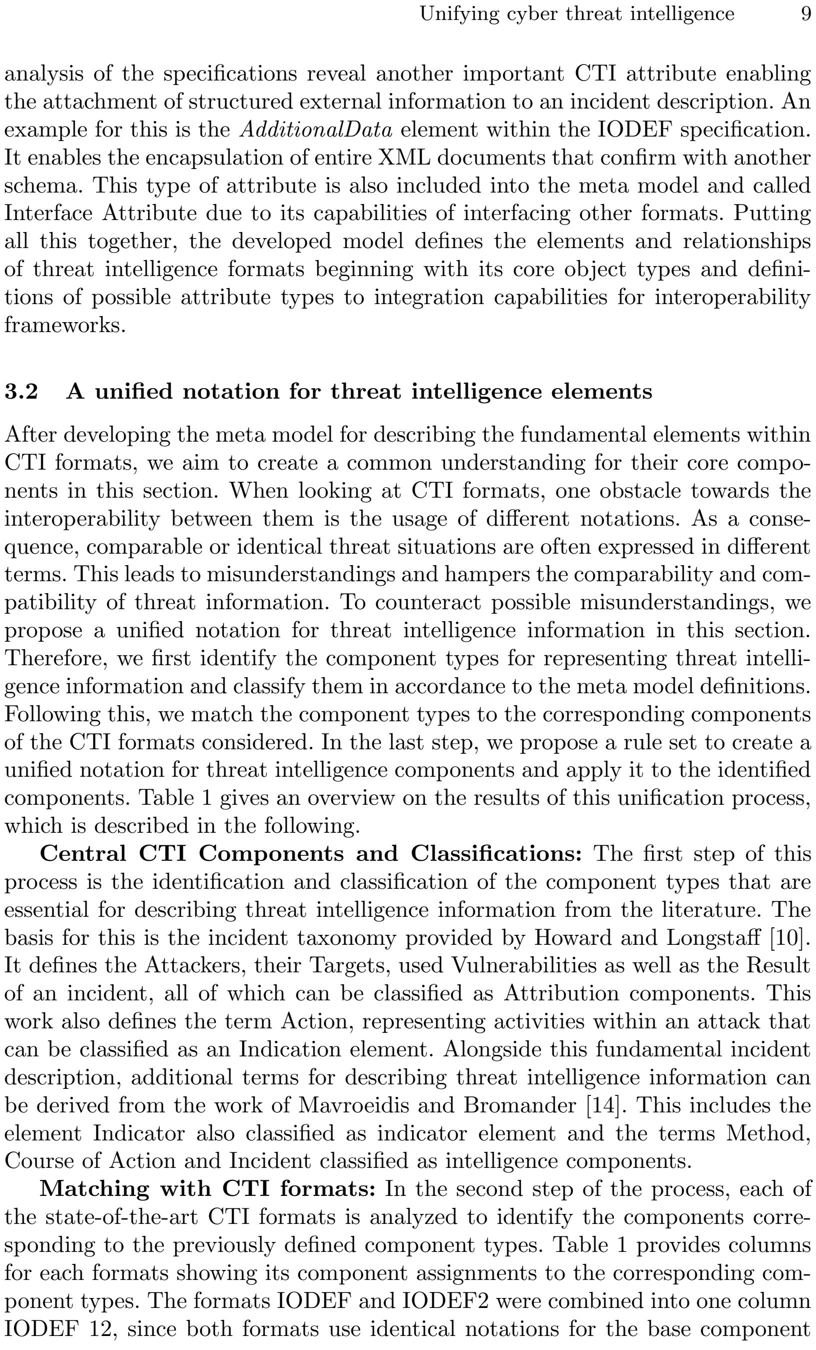
The specifications also confirm that the base elements of the formats considered can be categorized into Indicator, Intelligence and Attribution, while inheriting from the base Object. Vocabularies and Public Enumerations both of which inherit from the enumeration entity, show different usages within CTI formats. While Vocabularies find usage within any CTI object in different manifestations, Public Enumerations are restricted to the layers Intelligence and Attribution. The Intelligence layer can for example be provided with CAPEC[19], an enumeration for defining particular attack patterns, whereas the attribution layer can be provided with the previously described enumeration CVE for defining particular vulnerabilities within targeted entities. The Scoring Systems implemented within the examined formats are restricted to the attribution layer. More precisely, applied Scoring Systems such as CVSS and CWSS[20] describe the severity of vulnerabilities, whereas systems such as CCSS[21] describe the severity of configurations issues on specific targets. Since both targets and vulnerabilities are allocated to the Attribution layer, this also applies to Scoring Systems that describe their severities.

Finally, the CTI attribute definitions will be examined more closely. As stated above, a clear distinction between constrained and free-text attributes has to be made for a description of CTI formats. Considering the specifications, a great deal of both types of attribute can be identified. The considered formats define numerous free-text attributes such as description or notes allowing to insert arbitrary contents. Similarly, the formats also provide various constrained attributes such as *DateTime* enforcing a specific format. In addition to these types, the

[19] https://capec.mitre.org/

[20] https://cwe.mitre.org/cwss/cwss_v1.0.1.html

[21] https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir7502.pdf



analysis of the specifications reveal another important CTI attribute enabling the attachment of structured external information to an incident description. An example for this is the *AdditionalData* element within the IODEF specification. It enables the encapsulation of entire XML documents that confirm with another schema. This type of attribute is also included into the meta model and called Interface Attribute due to its capabilities of interfacing other formats. Putting all this together, the developed model defines the elements and relationships of threat intelligence formats beginning with its core object types and definitions of possible attribute types to integration capabilities for interoperability frameworks.

3.2 A unified notation for threat intelligence elements

After developing the meta model for describing the fundamental elements within CTI formats, we aim to create a common understanding for their core components in this section. When looking at CTI formats, one obstacle towards the interoperability between them is the usage of different notations. As a consequence, comparable or identical threat situations are often expressed in different terms. This leads to misunderstandings and hampers the comparability and compatibility of threat information. To counteract possible misunderstandings, we propose a unified notation for threat intelligence information in this section. Therefore, we first identify the component types for representing threat intelligence information and classify them in accordance to the meta model definitions. Following this, we match the component types to the corresponding components of the CTI formats considered. In the last step, we propose a rule set to create a unified notation for threat intelligence components and apply it to the identified components. Table 1 gives an overview on the results of this unification process, which is described in the following.

Central CTI Components and Classifications: The first step of this process is the identification and classification of the component types that are essential for describing threat intelligence information from the literature. The basis for this is the incident taxonomy provided by Howard and Longstaff [10]. It defines the Attackers, their Targets, used Vulnerabilities as well as the Result of an incident, all of which can be classified as Attribution components. This work also defines the term Action, representing activities within an attack that can be classified as an Indication element. Alongside this fundamental incident description, additional terms for describing threat intelligence information can be derived from the work of Mavroeidis and Bromander [14]. This includes the element Indicator also classified as indicator element and the terms Method, Course of Action and Incident classified as intelligence components.

Matching with CTI formats: In the second step of the process, each of the state-of-the-art CTI formats is analyzed to identify the components corresponding to the previously defined component types. Table 1 provides columns for each formats showing its component assignments to the corresponding component types. The formats IODEF and IODEF2 were combined into one column IODEF 12, since both formats use identical notations for the base component

Table 1. A unified notation for threat intelligence

Component type	Classification	STIX	STIX2	IODEF 1&2	VERIS	X-ARF	Rule	Mapping
Indicator	Indicator	Indicator	Indicator	Indicator	Indicator	-	2	Indicator
Action	Indicator	Observable	Observed Data	Record	Threat Action	Attachment	3	Action
Attacker	Attribution	ThreatActor	ThreatActor	ThreatActor	Actor	Source	1	Actor
Target	Attribution	Exploit Target	Exploit Target	System	Asset	Destination	1	Asset
Vulnerability	Attribution	Vulnerability	Vulnerability	Vulnerability	Vulnerability	Attachment	2	Vulnerability
Result	Attribution	ImpactAssessment	ImpactAssessment	Assessment	ImpactAssessment	-	1	Assessment
Campaign	Attribution	Campaign	Campaign	Campaign	Related Incidents	-	2	Campaign
Method	Intelligence	TTP	Attack Pattern	Attack Pattern	Vector	Category	1	Attack Pattern
Course of Action	Intelligence	Course of Action	Course of Action	Defined COA	Corrective Actions	-	2	Course of Action
Incident	Intelligence	Incident	Report	Incident	Incident	Incident	2	Incident

types. The assignment is done by semantically matching the component type definitions from the literature with the component specifications of the respective formats and is shortly described in the following. All assignments relate to components that allow clear allocations to the respective component types. Wherever no component type is available or the comparison results in incomplete or ambiguous matchings, the respective field is left blank.

A ruleset for creating a common CTI notation: In the last step, we propose a common notation for each of the components. To achieve this, we firstly propose a ruleset that defines how to derive notation elements from the previously matched components. The ruleset enables the reproducibility of notation elements as well as the derivation of notation elements for possible future extensions of the notation. The most important factor for this notation is full coverage for all characteristics of the CTI base components. According to this, the first rule defines that the component representing the according component type in the most general manner is mapped to the notation. This ensures a high degree of expressibility for the notation elements. Another important factor is the practicability of the notation, since the success of intelligence sharing formats depends on a widespread usage and therefore user acceptance. As a result, the second rule defines that, if a component notation is already used by different formats, it is mapped to the notation. Finally, the third rule defines that if none of these rules apply, the component type definition from the literature will be mapped to the notation. The defined rules are applied for the mapping of all CTI components in ascending order provided that the first matching rule determines which component will be mapped. An example for the mapping according to the first rule is the component type Target. Its mapping candidates are Exploit Target, System, Asset and Destination. Asset matches the first rule, since it is the most general representation of these candidates. It enables the description of systems, services etc. regardless of their role within the incident. In contrast to this, Exploit Target implies an assets role within an attack, whereas System and Destination only allow a limited view on affected entities. An example for the second rule is the attribute Campaign, which provides the mapping candidates Campaign and Related Incidents, both of which are equally general representations for the component type. As a result the second rule applies and Campaign is mapped as the attribute that is already used by multiple formats. Finally, Action is an example for the third rule. None of the candidates for Action provides a more general representation and none of them is used in multiple formats. Therefore, the component definition Action is mapped to the notation.

4 Towards unified threat intelligence data structures

In the previous section 3 we developed a meta model and a unified notation for the description of essential CTI elements. These findings are applied to develop a unified base model for the representation of CTI data formats in this section. The model illustrates the results of this work, contributes to the understanding of CTI data structures and serves as a basis for a discussion of future possibilities and challenges for CTI data formats. The unified CTI model is shown in figure 2 and described in the following. The essential components of CTI data formats defined in Chapter 3 are the model's foundation and have been translated into entity types for the development of this model. As a next step, the relationships between the entities are defined. In doing so, we refrain from consolidating redundant relationships, since the formats do not only represent data structures for the storage of information, but they may also be used as a tool for the expression incident information as for example shown in Böhm et al. [1]. A reduction of relationships also leads to reduced expression capabilities. Therefore, the integration of the relationships within the data model is achieved by obtaining existing relationships between entities from underlying formats and transferring them into the unified model. As a result, the relationships within the unified model are a superset of the relationships obtained from the underlying formats. The identified CTI entity categories Attribution, Intelligence and Indicator, as shown in chapter 3.1, are mapped as classification swimlanes to illustrate the entity classification assignments. In addition to the integration of entities, relationships and classifications, the entities are populated with structured properties according to the meta model. Therefore, vocabularies, enumerations and scoring systems

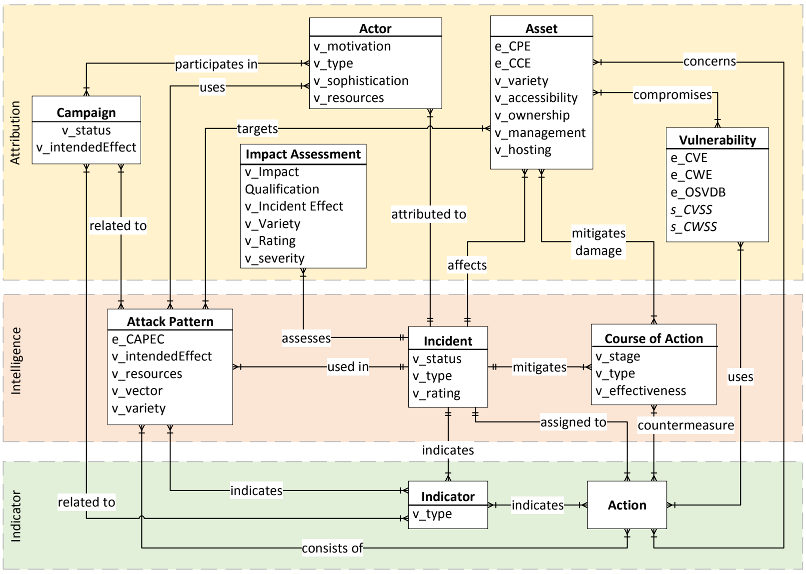


Fig. 2. Unified CTI data model

are assigned to the entities according to their occurrence within the CTI formats under consideration. Based on these properties, several entry points for possible improvements of CTI data formats are identified in the following and presented according to their classification lane.

Within the **Attribution Layer**, entity types *Actor* and *Campaign* only provide internal vocabularies for their structured description. External resources and scoring systems are not available for these types. As a result, properties and information about actors need to be collected and specified each time an incident is detected. An online resource, like an enumeration that collects information about attackers and allows to map it to an incident would be conceivable as a possible extension. *Asset* entities are described by vocabularies as well as by enumerations. One possible extension would be an additional scoring system to evaluate the criticality of the assets, as for example pointed out by Kim and Kang [13]. Moreover, the asset itself can be specified using the CPE[22] and CCE[23] enumerations, defining the software platform configuration and even allow the mapping of CVE vulnerabilities. However, since assets allow the relationship to single enumeration items, more complex systems with different components, such as cyber-physical systems, can hardly be described. *Impact Assessment* entities provide different internal vocabularies for the structured description of an assessment, as for example values for a subjective impact qualification rating from ”low” to ”high”. Although, results of impact assessments may vary widely across different companies, a common calculation base for a more informative exchange would be conceivable. This could for example be achieved using an impact assessment scoring system that allows the integration of environmental variables like the industry sector of a company and thus may provide a common calculation basis for the gravity of incidents. Within the **Intelligence Layer**, the entity *Course of Action* offers different possibilities for extensions. On the one hand, there are no metrics or scorings available, that would allow a transparent evaluation of the countermeasures conducted. Moreover, an external enumeration that for example provides known procedures for the treatment of specific incidents could contribute to the expressiveness of the course of action objects. Beyond that, the association of vulnerabilities with course of action entities could provide additional value. Similar to this, *Incident* and *Attack Pattern* entities, also do not allow the integration of scoring systems. The **Indicator Layer** lastly represents system observations without the use of contextual data. Therefore, enumerations or scoring systems are not available for these entity types, which the meta model has shown already.

5 Conclusion

In this paper we presented an approach for finding a common ground on describing CTI information. The developed meta model shows the syntax relation of various extant data formats to improve the structural understanding of CTI

[22] https://nvd.nist.gov/products/cpe
[23] https://nvd.nist.gov/config/cce/index

data formats. It also allows the distinction between enumerations and scoring systems for precise data representations and different attribute types that allow a certain degree of freedom when describing threat information. We also identified the key elements for CTI data formats, defined a ruleset for the creation of a unified notation to facilitate a common understanding for CTI elements. In the last step, the insights gained from the meta model and the unified notation were applied to create a unified base model for CTI data structures. Overall, the results of this work contribute to a common understanding for CTI data structures, serve as comparison tool for CTI formats and point out different future opportunities in this area. The developed meta model also represents a tool that allows the evaluation of CTI data format components and that allows to establish comparability between the components. It therefore represents one essential building block for future research, such as for the development of data quality metrics for CTI data. Ultimately, the results of this work support the creation of an industry standard for representing threat intelligence data. In this context, the developed rule set and the unified notation can be used as a basis for the integration of component definitions from different formats into one. The unified model serves as an initial model for the creation of a standardization for CTI data formats as well as for data format optimization within the integration process.

Acknowledgement

This research was supported by the Federal Ministry of Education and Research, Germany, as part of the BMBF DINGfest project (https:// dingfest.ur.de).

References

1. Böhm, F., Menges, F., Pernul, G.: Graph-based visual analytics for cyber threat intelligence. Cybersecurity **1**(1), 16 (Dec 2018)
2. Bourgue, R., Budd, J., Homola, J., Wlasenko, M., Kulawik, D.: Detect, share, protect. Tech. rep., ENISA (November 2013)
3. Brown, S., Gommers, J., Serrano, O.: From Cyber Security Information Sharing to Threat Management. Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security pp. 43–49 (2015)
4. Burger, E.W., Goodman, M.D., Kampanakis, P., Zhu, K.A.: Taxonomy model for cyber threat intelligence information exchange technologies. In: WISCS '14 Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security. vol. WISCS 14, pp. 51–60 (2014)
5. Falk, C.: An ontology for threat intelligence. 15th European Conference on Cyber Warfare and Security, ECCWS 2016 pp. 111–116 (2016)
6. Falk, C., Way, C.: Using an Ontology to Classify Cyber Threat Actors Using an Ontology to Classify Cyber Threat Actors (2018)
7. Fenz, S., Ekelhart, A.: Formalizing information security knowledge. Proceedings of the 4th International Symposium on Information, Computer, and Communications Security - ASIACCS '09 p. 183 (2009)

8. Fransen, F., Smulders, A., Kerkdijk, R.: Cyber security information exchange to gain insight into the effects of cyber threats and incidents. e & i Elektrotechnik und Informationstechnik **132**(2), 106–112 (2015)
9. Grecio, A., Bonacin, R., Nabuco, O., Afonso, V.M., De Geus, P.L., Jino, M.: Ontology for malware behavior: A core model proposal. Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE pp. 453–458 (2014)
10. Howard, J., Longstaff, T.: A common language for computer security incidents (1998)
11. Iannacone, M., Bohn, S., Nakamura, G., Gerth, J., Huffer, K., Bridges, R., Ferragut, E., Goodall, J.: Developing an Ontology for Cyber Security Knowledge Graphs. Proceedings of the 10th Annual Cyber and Information Security Research Conference on - CISR '15 (March 2017), 1–4 (2015)
12. Kampanakis, P.: Security automation and threat information-sharing options. IEEE Security & Privacy **12**(5), 42–51 (2014)
13. Kim, A., Kang, M.H.: Determining asset criticality for cyber defense. Tech. rep., Naval Research Lab Washington DC (2011)
14. Mavroeidis, V., Bromander, S.: Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. Proceedings - EISIC 2017 pp. 91–98 (2017)
15. Mavroeidis, V., Jøsang, A.: Data-driven threat hunting using sysmon. In: Proceedings of the 2nd International Conference on Cryptography, Security and Privacy. pp. 82–88. ACM (2018)
16. Menges, F., Pernul, G.: A comparative analysis of incident reporting formats. Computers & Security **73**, 87–101 (2018)
17. Obrst, L., Chase, P., Markeloff, R.: Developing an ontology of the cyber security domain. CEUR Workshop Proceedings **966**, 49–56 (2014)
18. Oltramari, A., Cranor, L.F., Walls, R.J., McDaniel, P.: Building an ontology of cyber security. CEUR Workshop Proceedings **1304**, 54–61 (2014)
19. Sillaber, C., Sauerwein, C., Mussmann, A., Breu, R.: Data Quality Challenges and Future Research Directions in Threat Intelligence Sharing Practice. Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security pp. 65–70 (2016)
20. Sprinkle, J., Rumpe, B., Vangheluwe, H., Karsai, G.: 3 Metamodelling. Model-Based Engineering of Embedded Real-Time Systems pp. 57–76 (2010)
21. Steinberger, J., Sperotto, A., Golling, M., Baier, H.: How to exchange security events? Overview and evaluation of formats and protocols. IEEE International Symposium on Integrated Network Management (IM) pp. 261–269 (2015)
22. Syed, Z., Padia, A., Finin, T., Mathews, L., Joshi, A.: Uco: A unified cybersecurity ontology. In: Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security. pp. 14–21 (2015)
23. Zhao, Y., Lang, B., Liu, M.: Ontology-based unified model for heterogeneous threat intelligence integration and sharing. In: 2017 11th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID). pp. 11–15 (Oct 2017)

3 Graph-based visual analytics for cyber threat intelligence

Current status:	Published
Journal:	Cybersecurity, Volume 1, Number 1, December 2018
Date of acceptance:	28 December 2018
Full citation:	Fabian Böhm, Florian Menges and Günther Pernul. Graph-based visual analytics for cyber threat intelligence *Cybersecurity 1(1): 16 (2018).*
Authors contributions:	Fabian Böhm 45% Florian Menges 45% Günther Pernul 10%

Journal Description: The Cybersecurity journal aims to systematically cover all essential aspects of cybersecurity, with a focus on reporting on cyberspace security issues, the latest research results, and real-world deployment of security technologies.

Böhm *et al. Cybersecurity* (2018) 1:16
https://doi.org/10.1186/s42400-018-0017-4

Cybersecurity

RESEARCH Open Access

Graph-based visual analytics for cyber threat intelligence

Fabian Böhm*, Florian Menges and Günther Pernul

Abstract

The ever-increasing amount of major security incidents has led to an emerging interest in cooperative approaches to encounter cyber threats. To enable cooperation in detecting and preventing attacks it is an inevitable necessity to have structured and standardized formats to describe an incident. Corresponding formats are complex and of an extensive nature as they are often designed for automated processing and exchange. These characteristics hamper the readability and, therefore, prevent humans from understanding the documented incident. This is a major problem since the success and effectiveness of any security measure rely heavily on the contribution of security experts.

To meet these shortcomings we propose a visual analytics concept enabling security experts to analyze and enrich semi-structured cyber threat intelligence information. Our approach combines an innovative way of persisting this data with an interactive visualization component to analyze and edit the threat information. We demonstrate the feasibility of our concept using the Structured Threat Information eXpression, the state-of-the-art format for reporting cyber security issues.

Keywords: Cyber threat intelligence, Visual analytics, Usable cybersecurity, STIX

Introduction

Over the last years the number of IT security incidents has been constantly increasing among companies. In order to keep pace with this development, there is a necessity for ever-improving protective measures. As single entities are no longer able to handle the vast amount of possible attack scenarios acting collaboratively against such attacks is an emerging trend. It is widely believed that cooperative approaches, in particular those based on the exchange of threat intelligence information, can contribute significantly to improve defensive capabilities (Shackleford 2015). A key factor for realizing cooperative approaches are the underlying threat intelligence data formats. They offer a semi-structured representation of identified threats and ensure a common understanding of security-related observations. As they document incidents using general mark-up languages, a common characteristic of these formats is a good machine-readability.

However, text-intensive and semi-structured data is of very little use for security experts due to its extent and lack of human-readability. This is a major problem when taking the role of security experts in today's companies into consideration. As the success and effectiveness of incident prevention, detection, and reaction rely heavily on the knowledge of security experts (Shackleford 2016; Luttgens et al. 2014), they need to understand what happened, how to react appropriately, and how to prevent new outbreaks of cyberattacks.

Structured threat intelligence is of great value for experts as it enables them to understand threats and attacks. However, this is only possible when experts are able to read and analyze this information. It is further crucial for experts to easily edit it in order to

* Correspondence: fabian.boehm@ur.de
Department of Information Systems, University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany

include any additional or missing information. The interaction requires an integrity-proof approach to persist original data in order to ensure the availability of untampered evidence for possible subsequent court cases.

We propose KAVAS, a knowledge-assisted visual analytics concept for the Structured Threat Information eXpression (STIX). KAVAS enables security experts to analyze and enrich cyber threat intelligence (CTI) data. We combine a novel way of persisting this semistructured data in a graph-based database with an interactive visualization. To demonstrate the feasibility of KAVAS we utilize the state-of-the-art format for structuring CTI information, STIX 2. Our work aims to improve the accessibility of cyber threat intelligence for security experts and to include them in the process of creating a comprehensive documentation for security incidents.

The remainder of this paper is structured as follows. Section 2 introduces the background of our work with regard to related research fields. In Section 3 we analyze related work and reach out for introducing the addressed research gap. This chapter is followed by the description of applied concepts and design decisions we made for KAVAS in Section 4. After introducing the main concepts of KAVAS we proceed to showcase how our approach works in Section 5. Section 6 qualitatively evaluates the applied approach to make threat intelligence accessible to security analysts. We conclude in Section 7 by discussing our concept and identifying future work.

Background

This section provides an overview of the Structured Threat Information eXpression format STIX, which is the state-of-the-art project for semi-structured representation of cyber threat intelligence information. Furthermore, a general view on knowledge and its role in the field of visual analytics is given.

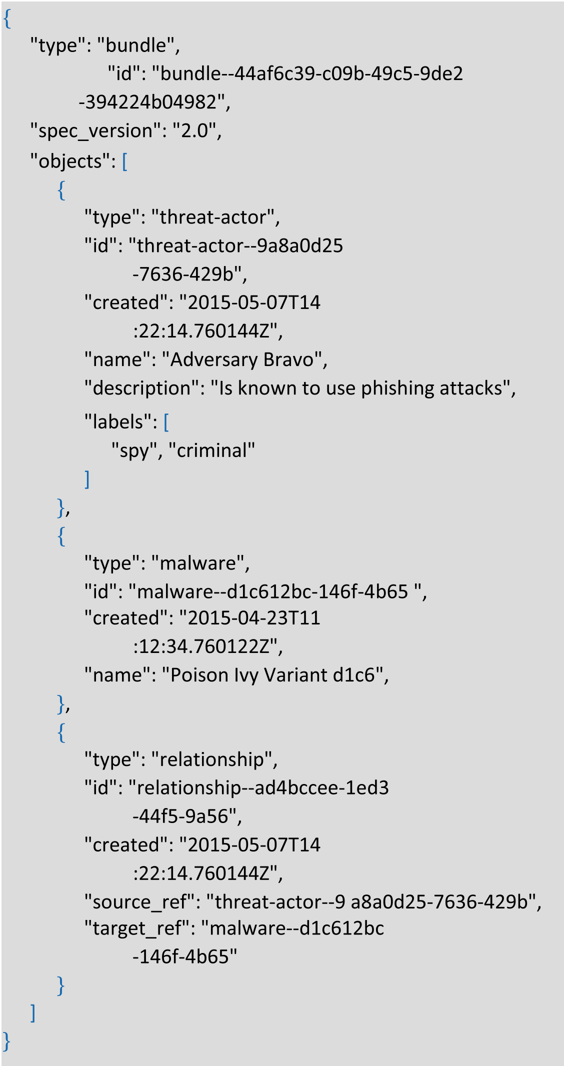
Structured threat information eXpression (STIX)

As argued above, structured formats are a key element within the threat intelligence exchange process because they pre-define which information can be shared. Additionally, these formats define requirements for the information density of the data to be shared. Depending on the specific use-case and the required contentual extent, the literature provides several formats that support structuring threat intelligence information. Examples for such formats are IODEF,[1] VERIS,[2] and STIX.[3] The primary focus of IODEF is the exchange of incident information between Computer Emergency Response Teams (CERTs), whereas VERIS focuses the measurement and management of risks involved in incidents. STIX 2, in contrast, is not bound to a specific use case and provides a comprehensive tool set for the representation of various information about incidents. As it is the format with the broadest possibilities in application (Menges and Pernul 2018), we focus our work on STIX 2 as the most recent version of STIX. This choice is further substantiated by STIX being the de-facto standard format for the exchange of threat intelligence information at present, which can also be anticipated for its successor STIX 2 in the near future (Shackleford 2015; Sauerwein et al. 2017). It provides the most extensive data structures among the available formats as shown by Asgarli et al. (Asgarli and Burger 2016) as well as by Menges and Pernul (Menges and Pernul 2018). This allows a wide ranging integration of expert knowledge into the analysis process. STIX 2 also provides highly flexible data structures allowing interactions of domain experts with very few limitations.

Regarding the content, STIX 2 provides a holistic representation for incident information, which is structured using the lightweight JavaScript Object Notation (JSON) file format. The data format provides two core component types: A STIX Domain Object (SDO) describing the characteristics of an incident and a STIX Relationship Object (SRO) describing relationships between those characteristics.

In its current version, STIX 2 specifies SDO elements for the representation of the attacking entity, event data describing the occurred incident as well as countermeasures initiated by the victim entity. The representation of the attacking entity includes information about the threat actor, the objectives, tools and attack patterns used within an attack. It also supports the description of entire attack campaigns and the attribution of attackers to such campaigns. The lateral movement of an incident can be represented using information such as exploited vulnerabilities, detected malware or digital identities involved in the incident. Actions taken to prevent an attack as well as responses to an attack can also be represented and associated to corresponding incidents afterwards.

Furthermore, STIX 2 specifies SRO elements to dynamically connect SDO elements. These connections can be realized using Relationship and Sighting Objects. Relationship objects indicate dependencies between SDOs, whereas Sighting objects refer to observed occurrences of SDOs. This allows building highly flexible representations for incidents only limited by the SDO definitions that are available within the data model (Piazza et al. 2017a; Piazza et al. 2017b). To encapsulate fully captured incidents, STIX 2 specifies an additional bundle element encapsulating all SDO and SRO elements captured in the course of an incident. Listing 1 gives a short example of a STIX 2 bundle.



```
{
   "type": "bundle",
            "id": "bundle--44af6c39-c09b-49c5-9de2
        -394224b04982",
   "spec_version": "2.0",
   "objects": [
      {
         "type": "threat-actor",
         "id": "threat-actor--9a8a0d25
              -7636-429b",
         "created": "2015-05-07T14
              :22:14.760144Z",
         "name": "Adversary Bravo",
         "description": "Is known to use phishing attacks",
         "labels": [
            "spy", "criminal"
         ]
      },
      {
         "type": "malware",
         "id": "malware--d1c612bc-146f-4b65 ",
         "created": "2015-04-23T11
              :12:34.760122Z",
         "name": "Poison Ivy Variant d1c6",
      },
      {
         "type": "relationship",
         "id": "relationship--ad4bccee-1ed3
              -44f5-9a56",
         "created": "2015-05-07T14
              :22:14.760144Z",
         "source_ref": "threat-actor--9 a8a0d25-7636-429b",
         "target_ref": "malware--d1c612bc
              -146f-4b65"
      }
   ]
}
```

Listing 1 Exemplary STIX 2 bundle

This listing shows the two SDO elements *threat-actor* and *malware* as well as the SRO element *relationship*, which connects the SDO elements using its properties *source ref* and *target _ref*. This example intends to illustrate the notation for objects and dependencies within the format as well as to give an impression of the possible complexity considering more extensive STIX 2 files.

Whenever the term "STIX" is used in the following sections, we actually refer to STIX 2.

Knowledge-assisted visual analytics

Visual Analytics (VA) is a combination of two important analytic reasoning processes: interactive visualization and automated analysis both striving to gain new insights (Keim et al. 2010). Keim et al. (Keim et al. 2008) define the creation of insight or knowledge as the final step in their widely accepted process for VA. This definition and other VA processes describe knowledge as a solely human artifact. However, not only humans own knowledge but a specific type of knowledge also exists for any automated analysis method included in VA (Fayyad et al. 2002; Sacha et al. 2014).

Therefore, knowledge-assisted visual analytics distinguishes the terms explicit and tacit knowledge (Nonaka and Takeuchi 1995; Polanyi 1983). Explicit knowledge can be defined as machine knowledge which can be read, processed, and stored by machines. Tacit knowledge is very specific to the individual and specialized as only humans are able to extract this knowledge type. In the context of knowledgeassisted visual analytics, tacit knowledge can be subdivided into smaller notions: 1) operational knowledge and 2) domain knowledge (Chen 2005). By having the appropriate operational knowledge a user knows how to interact with a visual analytics system. Domain or context knowledge is the ability of a user to interpret the visual representation regarding a specific context. Only a combination of these two types of knowledge enables users to understand the message told by a visual analytics system and thus to derive new knowledge (Chen 2005). Knowledge-assisted visual analytics aims to support the exchange of all these different knowledge types.

These exchanges can be formally described using knowledge conversion processes (Nonaka and Takeuchi 1995). Chen et al. (Chen et al. 2009) adapt these processes for information visualization. Wang et al. (Wang et al. 2009) as well as Federico et al. (Federico et al. 2017) further substantiate the concept of knowledge conversion to visual analytics with a special focus on explicit knowledge. The four conversion processes are namely: Internalization, Externalization, Combination, and Collaboration.

Internalization in knowledge-assisted visualization encompasses the transformation of explicit knowledge to tacit knowledge through visual interfaces. It supports humans in order to understand and transform explicit knowledge into domain knowledge (Wang et al. 2009). From a visualization perspective, this process is similar to the concepts of sensemaking (Pirolli and Card 2005) and insight or knowledge generation (Sacha et al. 2014; Chang et al. 2009). Internalization in terms of visualization can be described as follows: explicit knowledge is visually represented and through interactive exploration users gain tacit knowledge. Internalization is a high-level description of the generation of insight which is the primary goal and process of any visualization (Chen et al. 2009; Chang et al. 2009).

Externalization describes the transfer of knowledge along the opposite direction in contrast to internalization. It is a process where tacit knowledge is translated to explicit knowledge based on the insight of a user. There are existing prototypes in the visualization community showing that visualization tools taking externalization into consideration is suitable and effective for persisting and making use of experts' domain knowledge (Federico et al. 2017). Externalization can be applied using two main approaches. First, the more frequently applied approach is enabling users to directly transfer their knowledge. There exists a range of possibilities for implementing direct externalization. Examples are adjusting machine learning algorithms' parameters (Theron et al. 2017), adding patterns and rules to a knowledge database (Wagner et al. 2017) or changing an ontology used by automated analysis methods (Wang et al. 2009). Second, the other way to implement externalization is an implicit one by inferring explicit knowledge based on interactions of users with the visualization (Endert et al. 2012; Zhong et al. 2018). For example, dragging a node to a different location could be used to update and adjust the model of a clustering algorithm to fit the new position of the node.

Collaboration characterizes the exchange of tacit knowledge between humans (Wang et al. 2009). This process does not explicitly rely on computers and visualization as the most common form of sharing tacit knowledge is direct communication. However, collaboration can be supported through visual interfaces and the possibilities to externalize tacit knowledge and therefore, making it accessible for others at any time, supporting them to improve their own knowledge (Coleman et al. 1996).

Combination is a process where explicit knowledge from different sources is incorporated into an existing explicit knowledge system. It helps to improve available knowledge and to combine different bodies of explicit knowledge. This process is mostly independent from any visual representation of the explicit knowledge (Wang et al. 2009). However, users are integrated into this process by supporting the combination, identifying relations and finding inconsistencies or redundancies.

The development of knowledge-based interfaces and the representation of knowledge generated throughout the entire analytical process has been declared a key challenge for visual analytics research (Thomas and Cook 2005; Pike et al. 2009). However, in the domain of cyber security this is still underdeveloped.

Related work

Only few scientific publications tackle the problem of making threat intelligence information understandable for security experts by using visual interfaces. Even less work is available in the area of visual analytics systems specifically designed to display STIX.

Leichtnam et al. (Leichtnam et al. 2017) introduce a visualization approach for heterogeneous data sources. To transform the diverse data into a normalized model they derive a proprietary data model inspired by STIX. They build a visualization for their proprietary format. However, a visual representation for complex threat intelligence information documented with STIX itself is not provided.

A visualization displaying STIX in its full comprehensiveness is built by the STIX community itself.[4] This visualization builds a visual representation of a STIX bundle but lacks clear and structured design principles. Especially the functionality for security experts to convert their domain knowledge into machinereadable threat intelligence knowledge is missing.

While there is ongoing research in the area of structured formats for cyber threat intelligence (e.g. STIX) (Sauerwein et al. 2017) as well as knowledge-assisted visual analytics (Federico et al. 2017), there are, to the best of our knowledge, no efforts towards combining these two concepts in order to make threat intelligence information accessible for security experts.

In order to address this research gap, we define the following three requirements for our solution:

- **R1 - Handling complex threat intelligence data:** Enable integrity preserving storage and management of STIX as a notion of explicit knowledge in an appropriate database system rather than processing JSON files.
- **R2 - Visual representation of STIX:** Create an interactive visualization for STIX-based CTI information allowing security experts to derive knowledge and gain insights from an incident documentation.
- **R3 - Conversion of experts' knowledge:** Allow the exchange of explicit knowledge and security experts' tacit knowledge. Domain knowledge can be made available in the semi-structured STIX description of an incident by externalization. Therefore, the incident can be described more comprehensively and experts can benefit from each other's knowledge.

Our concept can be interpreted as a knowledge view in the information visualization framework introduced by Shrinivasan and van Wijk (Shrinivasan and van Wijk 2008) in 2008 to support analytical reasoning.

Concept and design

This section introduces the concept and design decisions made for the two main components of KAVAS: its persistence layer called Cyber Threat Intelligence Vault

(CTI Vault) to store and manage STIX as well as the corresponding visual analytics component to enable users to understand and interact with complex threat intelligence information. These concepts are aligned to the previously defined key requirements for KAVAS.

CTI vault

Hereinafter, we propose a concept for the persistence and handling of STIX cyber threat intelligence information.

R1 - handling complex threat intelligence data

STIX is designed as a graph-based model, which defines its domain objects as graph nodes and their relationships as edges. Therefore, we have chosen a graph database, as underlying technology in order to persist intelligence data appropriately.

> The CTI Vault serves as an extensible knowledge base, providing access for domain experts to the threat intelligence information, which can be seen as a notion of explicit knowledge. It represents a structured data storage for gathering captured incident data, which originate from individual files in JSON format. It serves as a technical foundation for storing incident information and additional domain expert knowledge, such as perceived similarities, differences and relationships between the different incidents.

Due to the dynamic data structures of STIX, the storage needs to provide capabilities for persisting data in a way that allows the integration of arbitrary relationships between the stored entities. Another essential requirement for the data storage is to assure integrity for the captured incident information. This is of special importance as the threat intelligence information could serve as piece of evidence in possible subsequent court cases. Therefore, it has to be ensured that interactions with domain experts will not distort any of the captured data, while preserving capabilities for enriching the captured data with additional information simultaneously.

To achieve these requirements, a differentiation between *inventory data* and *appended data* has to be made within the data storage. The inventory data, which represents the data foundation for incident information, describes all data that has been captured within an incident. The threat information contained in the stored entities as well as their relationships may not be changed after their initial storage and can consequently be considered constant. Therefore, this data has to be read-only. However, this is different for the use of appended data. These entities may be inserted, altered and deleted at any time and are intended to be connected with inventory data. Whenever information is edited, it has to be ensured that none of the operations performed on appended entities will influence the integrity of the inventory data.

The proposed concept is influenced both by the defined data structures within the STIX specification and the requirements for an interaction of domain experts with these data structures. However, the base requirement for the concept is the alignment to the STIX specification, to ensure the compatibility with the STIX data structures. This preserves the ability to exchange threat information with any endpoint compatible to STIX. Considering the requirements defined above, we firstly introduce an approach for persisting inventory data. This will be achieved by mapping the data available in the STIX data format, into a database representation.

The concept is subsequently extended by an approach for enriching the inventory data with appended data allowing the association of threat information to domain expert knowledge. Summarizing, the concept for handling complex threat intelligence data is based on the following two requirements, which will be specified in more detail afterwards.

- **R1.1 - Structured storage for threat intelligence data:** The collected data is stored in a structured way within a graph database as inventory data. The data storage has to be aligned to the STIX specification, allowing arbitrary relationships between the stored entities.
- **R1.2 - Integrity-proof storage and enrichment of persisted data:** A further requirement for the storage of threat intelligence data is to guarantee data integrity from insert operations onward. Moreover, subsequent update operations of the inventory data must not endanger its integrity. Therefore, it is mandatory to introduce a provenance process for every performed enrichment.

R1.1 - structured storage for threat intelligence data

To realize a concept of storing inventory data into the database, it is necessary to take a closer look at the STIX specification as well as to consider possibilities for the representation within a graph database.

The specification of STIX defines SDOs for the representation of threat intelligence information on the one hand and SROs defining relations between domain objects on the other hand. Both SDO and SRO are specified as stand-alone objects in STIX that allow to store multiple properties. According to the specification, SRO objects represent the relationships within the model by holding additional properties pointing to a source and target reference, each of which has to be a SDO. The combination of SDOs and SROs builds a directed graph, in which the first ones represent graph node objects and the latter ones represent edges connecting these nodes.

In contrast to this type of representation, graph databases allow the use of object types for creating nodes, whereas edges cannot be represented using object types. This leads to the necessity of adjusting the type of representation within the database in order to properly translate STIX into the database representation. Our approach for adjusting these interrelations between the STIX objects is visualized in Fig. 1(a) and described in more detail afterwards.

Our concept defines the representation of both SDO and SRO as physical nodes within the graph database. While SDOs act as self-sufficient nodes, SROs represent the relationships between SDOs and, therefore, act additionally as logical STIX edges. Finally, information about the source and target attributes of the relationship is transformed into physical edges realizing these relationships within the database. This leads to a representation that fully maintains the structural integrity of the STIX data model on the one hand and allows to map relationship properties into logical edges on the other hand. Conclusively, this results in a logical representation for the directed-graph structure of STIX, which is stored using a physical non-directed graph structure within the database.

In addition to this, the STIX specification defines detected incident information to be pooled in relation to a root *bundle* element. Since the physical graph is non-directed, the bundle element can be connected to every SDO contained within an incident to achieve the pooling. This element can serve as an entry point for the traversal of incident information at the same time.

R1.2 - integrity-proof storage and enrichment of persisted data

Within the process of storing data into the CTI Vault, the integrity of captured data is essential to preserve its evidential significance for any subsequent forensic analyses or even for court cases. The proposed concept provides two different mechanisms to guarantee the integrity for stored incident information. On the one hand, the integrity of incident information has to be ensured when it enters the system for the first time, on the other hand, changes on persisted information have to be conducted in an integrity preserving manner. The integrity of inserted information is preserved using controlled redundancies. Inserted information will intentionally not be checked for redundancies to prevent any possible distortion of this data. The insertion of redundant data is possible, since the graph database assigns an internal unique identifier for every element inserted. This, in turn, prevents objects with the same content from producing collisions. However, delimitation for redundant

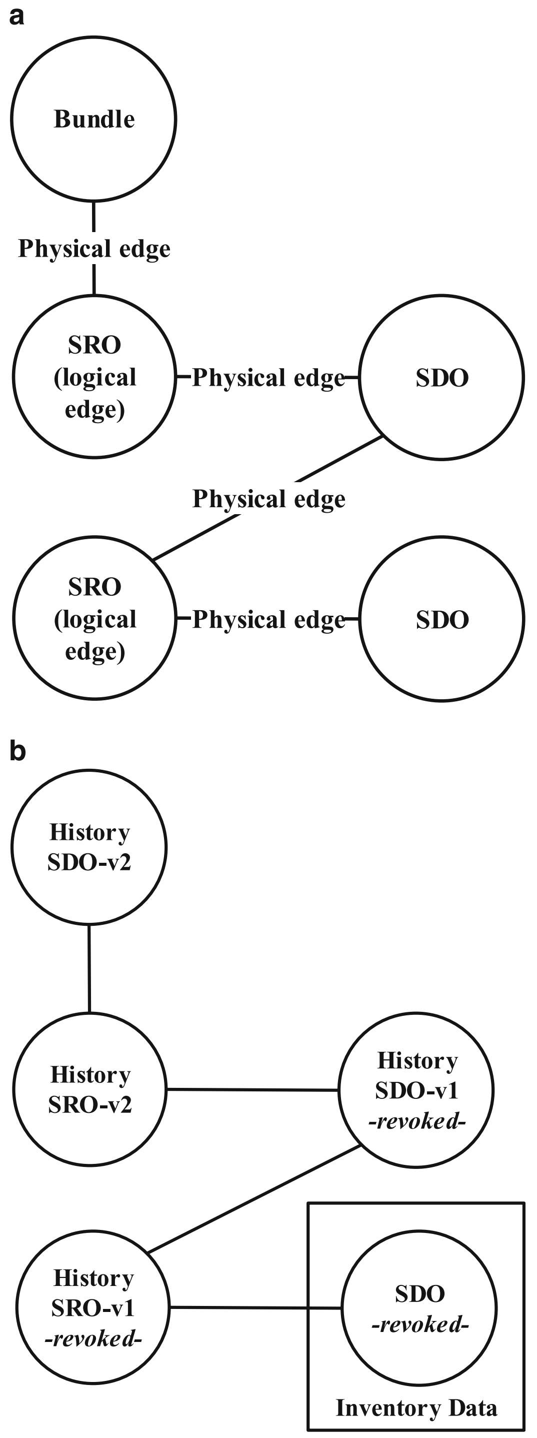


Fig. 1 CTI Vault of KAVAS. **a** Persisted Inventory Data (**b**) Enriched Inventory Data

objects remains still possible due to the pooling of elements and their affiliation to their root element, namely their bundle. The only exception for this are insertions of redundant elements within one bundle. However, this would only be the case if the elements contain identical STIX unique identifiers, which makes them both syntactically and semantically identical and consequently leads to a unification of these elements.

In addition to the concept of integrity-proof persisting for inventory data, the CTI Vault is designed to provide capabilities to store additional data that enriches the available information with domain knowledge of experts. Therefore, it needs to enable the extension of existing objects and relationships of inventory data. Since the enrichment of data with domain expert knowledge is not necessarily a singular event, the database also needs to provide capabilities for historicization of all performed changes.

As stated above, the concept of enriching inventory data is based on two main requirements. It has to be ensured that the inventory data will not be altered at any time and that the enriched data is still fully compatible to the STIX 2 specification. Consequently, the concept for enriching inventory data is also based on the STIX data structures.

According to this, only valid SDO or SRO elements that meet the STIX specification may be appended to the inventory data. Similar to the persistence of inventory data, appended data is also structured based on SDO nodes that are connected using logical and physical edges respectively. This results in a consistent database structure.

Figure 1(b) shows an exemplary SDO element within the inventory data extended by two subsequent changes, which are realized using a versioning structure within the database. In this process, supplementary nodes are added for each change. To indicate that nodes have been overwritten, the CTI Vault flags the respective former versions as “revoked” according to the STIX specification (Piazza et al. 2017b).

The first change is realized by creating a version SDO-v1 that extends the information within the original SDO, which is part of the inventory data. SDO-v1 in turn is connected to its base entity using a newly created relationship object SRO-v1. The second change is realized by creating a further version SDO-v2 and a corresponding relationship SRO-v2. It is important to maintain the order of succession for all changes performed. As a result, this concept enables every node within the inventory data to carry its own chain of edited data.

The presented concept for persisting cyber threat intelligence information in the STIX format fulfills therefore our requirement **R1**. This concept is the basis to support the *Combination* process as we interpret the STIX information stored in the CTI Vault to be explicit knowledge (Chen et al. 2009; Ackoff 1989).

Visualization design

The visual analytics component enables security experts to analyze, understand, and edit threat intelligence information. As described in Section 2.1, STIX is a powerful but text-intensive and semi-structured threat intelligence format. A single bundle can easily reach thousands of lines for complex incidents. This makes the documentation very hard to analyze and understand for security experts. This gets even worse when an expert appends information to the STIX file. In order to externalize domain knowledge, the complex structure of the format including all possible objects, relationships, their attributes, and allowed values for the attributes has to be known. To support the tasks of analyzing and enriching threat intelligence documented in STIX, we developed a visual analytics component on top of the previously introduced CTI vault.

Figure 2 shows the visualization component in the overall context of the system and defines the relations between KAVAS and security experts: the visualization uses the explicit knowledge stored in the CTI vault and maps this knowledge into an interactive view using the specification. The security experts can perceive the displayed knowledge to gain insight and situational awareness (Yen et al. 2014). At the same time they can use their operational knowledge to interact with the visualization in order to adjust the view specification or to enrich the information stored in the CTI vault.

R2 - visual representation of STIX

As STIX is designed to be a connected and directed graph of nodes and edges we are using a directed node-link diagram to represent knowledge persisted in the CTI Vault (Piazza et al. 2017a). This visualization technique is well suited for understanding threat intelligence as it reveals interconnections using nodes and edges (Severino 2018; Heer et al. 2010). Revealing the relationship between specific nodes (e.g. threat actors, used attack patterns and the targeted entities) is a crucial task of experts analyzing STIX. This makes the node-link diagram appropriate for the data structure at hand. However, Marty (Marty 2009) as well as Card et al. (Card et al. 1999) identify two main challenges when using node-link diagrams. To address those and to ensure the design of a suitable visual representation of STIX, we need to fulfill the following more specific requirements:

- **R2.1 - Render complex threat intelligence:** The cyber threat intelligence persisted in the IoC Vault

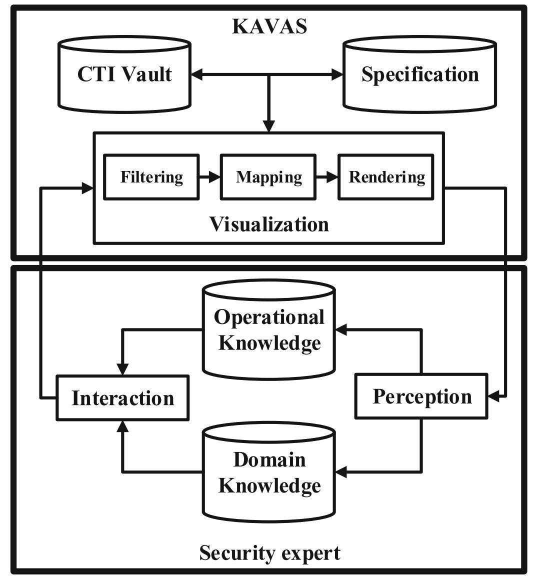


Fig. 2 High-level structure of KAVAS

is displayed in a suitable visual representation. The visual representation is fully capable to parse, map, and render all information provided in STIX bundles according to the STIX specification.

- **R2.2 - Scalable visual display:** As STIX bundles can contain hundreds of objects and even more links between these objects, the visual display has to be scalable. This can be assured by an appropriate layout algorithm and interactions for the users to adjust the layout.
- **R2.3 - Exploratory analysis:** To allow that users can deduce tacit knowledge from the displayed explicit knowledge, the visual representation must provide interactions supporting the analytical process of users.

R2.1 - render complex threat intelligence

The first challenge is to identify an appropriate way for positioning the nodes and links in the visualization space. SDO and SRO are abstract data constructs and do not have any natural position like on a geographical map. The InfoVis pipeline introduced by Card et al. (Card et al. 1999) is a process for creating views based on abstract data. By applying this process to SDOs and SROs, we are able to generate a visual representation of STIX. The following paragraphs describe our adaption of the InfoVis pipeline.

Originally, the pipeline starts with a data analysis responsible for data cleansing or interpolating missing values. We omit this step in our visual component as the CTI Vault is designed to persist only semantically and syntactically correct STIX bundles. Therefore, our view generation process starts with filtering the data to be visualized, as shown in the *Visualization*-box in Fig. 2. *Filtering* is realized by receiving a single userselected STIX bundle from the vault. This ensures that the analyst only sees information related to the bundle of interest. According to the InfoVis pipeline this single STIX bundle and the corresponding objects are referred to as focus data (Card et al. 1999).

The STIX objects in the focus data do not have any available positioning in the visualization space yet. Therefore, we need to transform the STIX-specific data structure into displayable nodes and edges in a mapping-step. As the STIX format defines SDOs to be nodes and SROs to be links in its graph-based structure, we adopt this definition. However, we had to make adjustments to improve the comprehensibility of a visually represented STIX bundle. We are displaying not only SROs as links in the node-link diagram but also important relationships embedded into SDOs referencing other objects. These embedded relationships are important to understand underlying connections in the threat intelligence information. For example, when an incident report is documented with STIX, embedded relationships of the report highlight which objects the report refers to. This and similar information can be important to an expert when analyzing an incident. To allow a fast perception of embedded links, we decided to include embedded relationships of SDOs as specially denoted edges into the diagram.

Additionally, we had to adjust the way STIX Sighting objects are represented in our visualization to retain a visually understandable way of representing STIX. These objects denote the insight that an attack, threat actor, campaign or other domain object was seen (Piazza et al. 2017b). They are used whenever an already documented attack is identified at another entity as well. Therefore, they are applied to track who was targeted as well as which attacks were performed. A Sighting object is specified to be a relationship. This means it would appear as a link in the visual representation although a Sighting is only connected to other SDOs via embedded relationships. We decided to include Sightings as nodes which are connected to SDOs via their different embedded relationships in the visual STIX representation to improve the perception of Sightings. These design decisions enable rendering all STIX objects as nodes and links on the canvas.

R2.2 - scalable visual display

Another issue of node-link diagrams is their limited scalability in terms of large numbers of highly connected nodes. They tend to resemble hairballs which makes it hard for users to understand the displayed information.

STIX bundles with large numbers of SDOs and SROs hamper a fast visual perception of relationships between the objects. However, a well-chosen layout algorithm and interactive functionalities for experts to adjust the layout can reduce this problem (Marty 2009). These functionalities are of great importance to ensure that a user is able to customize the visual representation of the STIX bundle. To arrange the information appropriately on the visualization canvas we apply a force-directed graph layout (Kobourov 2010). This algorithm creates a node-link diagram driven by different forces (e.g. gravity of node clusters, strength of links), which avoids overlapping as far as possible. However, due to the possible size and complexity of highly-interconnected STIX incident representations, it is necessary to provide interactive functionalities for security experts to adjust the layout themselves. This is especially necessary, when the automated force-directed algorithm is not capable to render a feasible layout anymore. In KAVAS we implement interactions allowing users to drag and drop single nodes and pin them to the desired position. Additionally, users can browse into specific parts of the STIX bundle by zooming. If the amount of nodes is overstraining the user, filters can be applied to show and hide the different types of SDOs and SROs.

R2.3 - exploratory analysis

Our concept allows security experts to interactively explore visually represented incident documentation. This exploratory analysis follows the Information Seeking Mantra defined by Shneiderman: “Overview first, zoom and filter, details on demand” (Shneiderman 1996). The *Overview* is provided by the initially generated node-link diagram based on the STIX intelligence information. With common interaction patterns like Pan-and-Zoom, hovering actions, filtering and Drag and Drop, security experts can adjust the view (Heer and Shneiderman 2012). This fulfills the *Zoom and filter*requirement of Shneiderman’s mantra. *Details on demand* are displayed when an element of the node-link diagram is selected. By analyzing the visual STIX representation users broaden both their operational knowledge and their domain knowledge (Chen et al. 2009).

By implementing **R2.1**, **R2.2**, and **R2.3** in our approach, we are able to provide an interactive visual representation of the explicit knowledge embedded in the threat intelligence.

R3 - conversion of experts’ knowledge

KAVAS allows the enrichment and editing of cyber threat intelligence while preserving the integrity of the original information at the same time. The enrichment and editing is necessary to externalize any additional or missing information from the user’s domain knowledge. Preserving the integrity throughout this editing action allows the intelligence to serve as piece of evidence. In our approach, security experts are able to externalize their domain knowledge either through changing the attributes of existing SDOs and SROs or through adding new nodes and links. This functionality covers the *Externalization* process as users are able to transfer their domain knowledge to the CTI Vault, where it is preserved as explicit knowledge.

Our concept supports the *Collaboration* of several security experts by transforming it to explicit knowledge. This explicit knowledge can then be displayed to other users, which could further support them in their analysis of the incident. Thus, experts editing existing intelligence implicitly make their domain knowledge accessible for other users.

Visualization architecture

We adopted the classical Model-View-Controller (MVC) design pattern for the visual analytics component (Krasner and Pope 2000). This divides the application into three main interconnected parts to separate the internal representation of information and business logic from the visual presentation to a user. Figure 3 shows a high-level view on the MVC structure of the KAVAS visualization component. The MVC-structure of KAVAS shown in the figure is also aligned with the different steps of the InfoVis pipeline described earlier.

The *Database Connector* is the interface towards the available web services of the CTI vault enabling the visualization to retrieve threat intelligence data. It also enables the visualization to send updates to the database in case a security expert edited the STIX documentation. The visualization exchanges STIX-based documentations in JSON format with the vault.

The *STIX Parser* receives the JSON file from the *Database Connector*. It is responsible for parsing the file into instances of the SRO and the SDO data models. Both these models inherit a number of common properties every STIX object must contain. The models are specified in accordance with the STIX 2 specification (Piazza et al. 2017b). In addition to the simple attribute values, our models define the data type of the property and a description for the properties. They also define whether a property is required. All this information is extracted from the STIX specification to be able to parse CTI information from the vault and to create valid STIX documentations based on changes made by security experts. The model instances are held by the parser in two different lists; one containing relationship objects and the other containing domain objects. Parsing JSON into object instances has two main advantages: easy mapping and rendering of objects into a node-link diagram as well as assuring compliance of processed STIX objects with the specification.

As pointed out earlier the abstract STIX data has no position in the diagram yet. The *STIX Mapper* maps the parsed STIX objects onto the visualization canvas. It wraps every instance of the beforehand described STIX models with a *NodeType* or *LinkType*. These data models contain additional properties (e.g. position, movement speed, etc.) to enable the *NodeLink Controller* to render the *NodeLink View*, which displays the interactive visual STIX representation. The *View Specification* tells the *NodeLink Controller* important settings such as the current zoom factor, gravity, link length, node radius and others.

The details of any STIX object can now be shown by handing over the selected *NodeType* to the *ObjectDetails Controller*. This controller then queries the object lists of the *STIX Parser* to receive the corresponding STIX object instance. This instance is forwarded to the *ObjectDetails View* for displaying details-on-demand. When an expert edits the STIX description, the parser receives the changes from the controllers, changes the model if necessary and forwards the changes through to the *Database Connector* to the CTI vault.

Prototype

In the following paragraphs we explain applied technologies for implementing KAVAS and give some detail of its functionalities with a short and small-scaled working exemplary bundle. A prototype of KAVAS is available here: http://bit.ly/2v9mSna (Sauerwein et al. 2017). Please note that KAVAS is currently an academic prototype. The linked version serves as a proof of concept. We are aware of required improvements to allow the operative use of KAVAS. The most emergent improvements are scoped for further versions of KAVAS and are described at the end of this article.

Applied technologies

The KAVAS visual analytics component is exclusively based on open-source web technologies forming a client-server web application in combination with the CTI vault (see Fig. 4). The CTI vault serves as back-end, providing the underlying data storage as described in Section 4.1 in combination with an API that enables data access for the front-end application. The vault is realized using the Java-based graph database Neo4j (Asgarli and Burger 2016) as base technology. Consequently, we also chose Java as language for realizing the access to the database as well as the related business logic managing the access. This layer assures the compliance to the object constraints predetermined by STIX, such as the specified object definitions and relationships. This is necessary, since the graph database does not provide such capabilities. In order to provide web-based access to the storage application, the actual Java implementation is running on a JavaEE[5] based application server. This allows us to provide REST webservices that can be accessed from the front-end application. The main technologies on the front-end are Angular.io[6] and Angular Material[7] which are frameworks on top of

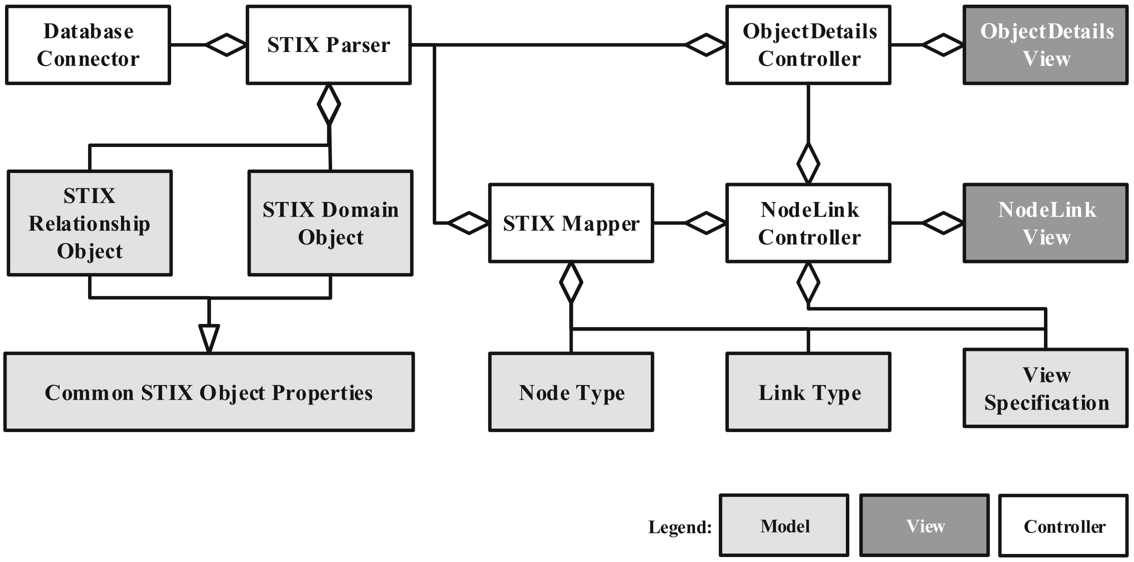


Fig. 3 MVC-based architecture of KAVAS' visualization

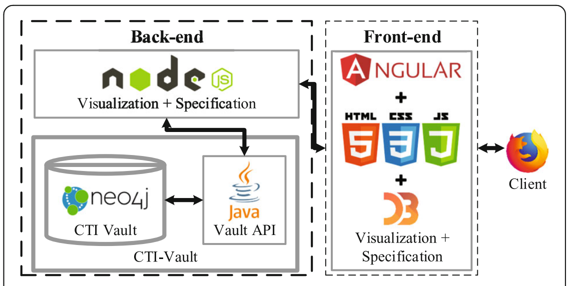


Fig. 4 Technologies used in KAVAS

HTML5, CSS3 and JavaScript. The interactive node-link diagram is implemented using the D3.js[8]-library.

Interactive exploration

Figure 5 displays different views of the visual interface of KAVAS. The bundle shown in the figure is part of the official example data sets for STIX 2.[9] Figure 5(a) shows the main view of the KAVAS visualization: an overview of a STIX bundle displayed as node-link diagram. The bundle itself documents an advanced persistent threat targeting the *Branistan Peoples Party (BPP)* which is one of the political parties of Branistan, a fictional country. The BPP's homepage is hit by an attack named *Operation Bran Flakes* where adversaries deploy *Content Spoofing* trying to insert false information into the BPP's web page. The campaign is rolled out by a *Fake BPP* which is most certainly sponsored by the *Franistan Intelligence* service, whereby Franistan is considered another fictional country. *The MITRE Corporation* detected and documented the attack.

An expert gets an overview (see Fig. 5(a)) of the STIX description in the node-link diagram after selecting the STIX bundle in the tool-bar's drop-down menu. The selected bundle is then received from the CTI vault, parsed and transformed for the visual display. To get a first glance of the documented incident, the expert can Pan-and-Zoom the diagram as well as drag and pin nodes to a fixed location on the canvas. Panning and zooming allows for interactive exploration. Dragging nodes across the canvas and pinning them to specific locations helps the analyst with adjusting the node-link diagram to be well arranged even for large numbers of nodes and edges. Whenever the mouse is moved over a node, KAVAS highlights the nearest neighbors of this node (see Fig. 5(b)). With enabling experts to select a node or link of the diagram and displaying the detailed properties of this STIX object (see Fig. 5(c)), KAVAS fully implements the Information Seeking Mantra for threat intelligence information.

Embedded relationships are not displayed as separated edges in Fig. 5(a). This is another functionality implemented in the visualization component. As described earlier, we map the embedded relationships of STIX objects as specially denoted edges. However, displaying all embedded relationships leads to incomprehensible diagrams very fast. Therefore, the embedded links as well as all other node or link types can be hidden or displayed interactively by the user.

Editing and enriching CTI information

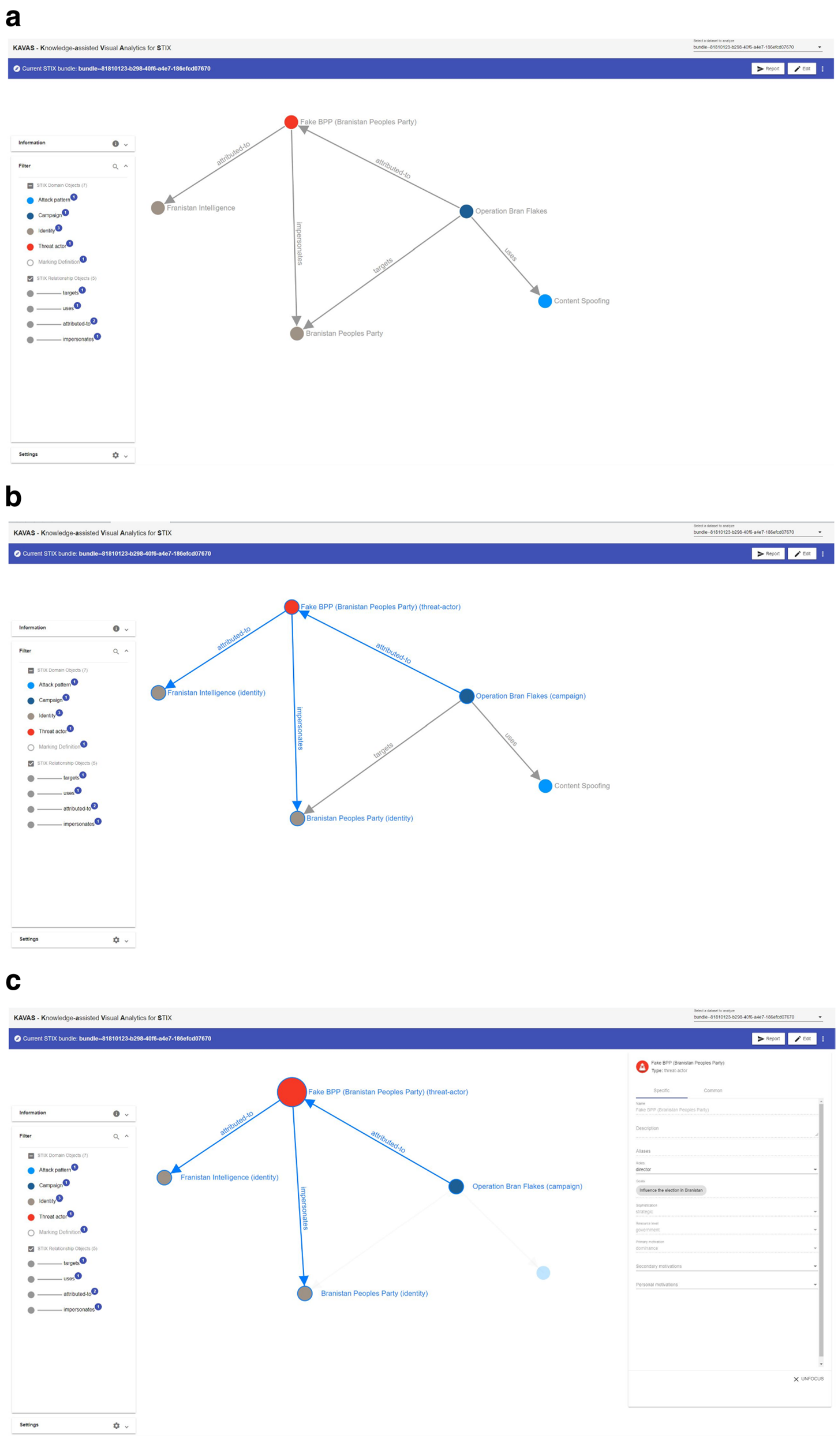
Analysts can enable editing whenever they want to change or add any information to the bundle. When this mode is activated the view itself stays the same to keep the analyst in the existing visual metaphor. However, the interaction behavior is different. Clicking on the blank canvas now triggers the process for adding a node to the diagram. The first step in this process is selecting the STIX object type as it defines the properties of the SDO. KAVAS adds the node to the SDO list in the STIX parser and displays it on the canvas. Afterwards, the tool brings up the details-on-demand window and the user can now edit the information for the newly added object.

Instead of dragging a node as described earlier, clicking and moving the mouse with a node triggers the process of adding an SRO while in *Editing*-mode. If the mouse is released on a node, a new edge, with the starting node as source and the ending node as target is added to the canvas. From here on, the process for adding the SRO to the parser and the canvas is similar to adding a SDO. Finally, the user sees the newly created link highlighted and the editable details-ondemand window.

By clicking an existing node or link in editing mode the properties of this STIX object can be changed except for some properties, which by definition should not be element to any changes throughout the whole life-cycle of an object (e.g. its ID).

After the user clicks to save in the details-on-demand window, the input is checked for its conformity with the STIX specification. If the object is conform it is parsed into a compliant JSON. This happens regardless of whether a new object is added or an existing one is changed. Afterwards the JSON is sent to the CTI vault where the data is persisted.

When an expert starts editing a STIX bundle, this specific bundle is locked in the IoC Vault. Other users can still load the bundle from the vault to analyze the corresponding node-link diagram. However, they cannot switch to editing mode and they are notified that the bundle is currently edited by another user if they try to edit the bundle. When the editing user finishes the work on the bundle or closes the browser, the bundle is unlocked in the vault. This is possible as changes to the bundle are only possible on the level of SDOs and SROs which have to be saved separately after they were


a
KAVAS - Knowledge-assisted Visual Analytics for STIX
Select a dataset to analyze
bundle--81810123-b298-40f6-a4e7-186efcd07670
Current STIX bundle: bundle--81810123-b298-40f6-a4e7-186efcd07670
Report
Edit
Information
Filter
STIX Domain Objects (7)
Attack pattern 1
Campaign 1
Identity 3
Threat actor 1
Marking Definition 1
STIX Relationship Objects (5)
targets 1
uses 1
attributed-to 2
impersonates 1
Settings
Fake BPP (Branistan Peoples Party)
attributed-to
Franistan Intelligence
attributed-to
Operation Bran Flakes
impersonates
targets
uses
Content Spoofing
Branistan Peoples Party
b
Fake BPP (Branistan Peoples Party) (threat-actor)
Franistan Intelligence (identity)
Operation Bran Flakes (campaign)
Branistan Peoples Party (identity)
c
Specific
Common
UNFOCUS

Fake BPP (Branistan Peoples Party)
Type: threat-actor

Name: Fake BPP (Branistan Peoples Party)
Description
Aliases
Roles: director
Goals: Influence the election in Branistan
Sophistication: strategic
Resource level: government
Primary motivation: dominance
Secondary motivations
Personal motivations

Fig. 5 Phases of Information Seeking Mantra in KAVAS A prototype of KAVAS is available at http://bit.ly/2v9mSna. The displayed bundle's id is *bundle–81,810,123-b298-40f6-a4e7-186efcd07670* and it can be accessed via the drop-down menu in the toolbar of KAVAS. **a** Overview (**b**) Zoom and Filter (**c**) Details on Demand

changed. Other users are now notified that the bundle is not locked anymore. When they activate the editing mode, the bundle is reloaded from the vault to ensure that they are working on the most recent version. They also can reload the bundle manually without switching the mode of action when they do not want to edit anything but still want to analyze the latest version of the STIX bundle.

Embedded knowledge processes

The KAVAS prototype is designed and implemented after a knowledge-assisted visualization approach. Therefore, the four knowledge conversion processes can be clearly identified within KAVAS' functionalities:

- *Internalization:* This knowledge conversion process describes the transfer of explicit knowledge into tacit knowledge through visual interfaces supporting humans to understand the explicit knowledge. KAVAS provides an interactive visual representation of explicit knowledge encompassed in the threat intelligence. In our system, internalization mainly happens through the interactive exploration of users. The node-link diagram and interaction functionalities aligned with the Information Seeking Mantra help users to inspect the knowledge and further support the discovery of unknown relationships and patterns which can become new domain knowledge.
- *Externalization:* Our concept allows tacit knowledge of domain experts to be externalized and persisted as explicit knowledge. Users can insert domain knowledge that does not yet exist in the threat intelligence information. Regardless of where the missing domain knowledge is originating, once acquired by the user, it can be directly inserted into the STIX bundle to augment threat intelligence. KAVAS allows this process through implementing means for users to directly edit the displayed STIX objects or add missing ones. Newly added information is persisted in the CTI Vault. After previously existing intelligence is changed, the original information is kept and linked to the updated version to ensure traceability of any changes to the STIX bundle.
- *Collaboration:* This process emerges when a user analyzes intelligence, which contains the externalized knowledge of other users. All available STIX information is persisted in the central CTI Vault and all intelligence displayed to the users is retrieved from this central intelligence storage. When one domain expert changes an incident description by editing existing intelligence or adding new pieces of information, this externalized knowledge is available for all other experts. Accordingly, having the CTI Vault as a centralized storage structure for all STIX intelligence and enabling users to externalize their domain knowledge, KAVAS supports the collaborative generation of tacit knowledge among its users.
- *Combination:* This process encompasses the insertion of new explicit knowledge into our existing knowledge base (CTI Vault), which is able to process any valid STIX bundle and to persist it. As a first step, it is highly important that the original bundle is stored regardless whether its information elements overlap with existing bundles. Hence, the bundle can be held in its original form and remains useful as possible evidence in court. After the initial storage of the original intelligence, further measures can be applied to detect and remove inconsistencies or redundancies. Currently, those measures are not yet part of the CTI Vault. However, the combination of existing explicit knowledge with new knowledge can be realized with our concept of the CTI Vault.

Evaluation

To validate our prototypical implementation of KAVAS and to provide first evidence of its usability and suitability to support knowledge conversion, we followed a two-fold research approach. An anonymous analyst survey validates the general suitability of the visualization approach for the addressed problem and eliminates usability issues of the interface. The survey is followed by expert interviews to confirm that KAVAS can facilitate knowledge conversions between domain experts and cyber threat intelligence.

Analyst survey

This survey intends to validate the relevance of the initial problem and the suitability of our design approach. Although, the survey cannot validate that the visualization facilitates all four knowledge conversion processes, it provides some hints whether the process of internalization is appropriately tackled.

Participants

The survey involved twelve security analysts from different academic institutions and companies such as internet service providers and security consultancies. The participants have a general understanding of threat intelligence. However, none of them is currently working with structured formats like STIX.

Design & Procedure

Staheli et al. (Staheli et al. 2014) propose a set of different aspects to evaluate visualizations for cyber security. Many of these aspects would need a more thorough user study. However, our survey is meant to give a first indication on the suitability of KAVAS for making cyber threat intelligence accessible for human analysts. Based on the definitions proposed by Staheli et al. (Staheli et al. 2014) we assess the dimensions *User experience*, *Usability and Learnability*, *Insight generation*, and *Feature set utility*. The questionnaire encloses questions with informal character. Nevertheless, all questions are answered on an interval Likert scale ranging from 1 to 5 with the first and last numerical value being labeled with a textual description indicating the scale from *1: not at all* to *5: quite a lot*. The questionnaire includes the following five questions:

- **Q1:** Is the analysis and understanding of incidents relevant for your company/institution?
- **Q2:** Is the proposed visual tool effective for an investigation of threat intelligence information?
- **Q3:** Is the proposed visual tool clear and understandable?
- **Q4:** Is the proposed visual tool adequate to display and enrich the available incident information?
- **Q5:** Does the tool overall help to understand what happened during the described incident?

An additional open field allows participants to report any further comments or suggestions on the tool.

Before the beginning of the survey, the analysts are introduced to the tool, its features and our motivation to build it. Subsequently, a JSON representation of a synthetic incident as described in Section 5.2 is shown. By using the JSON representation we are able to highlight the main problem with STIX-based intelligence, which is the low readability and accessibility of the format. Afterwards, the participants explore the incident freely and are asked to fill out the questionnaire.

Results

Considering Fig. 6 and Table 1 we derive the fact that the addressed problem is relevant for the respective company or institution of the analysts. The high standard deviation leads to the conclusion that the need for sharing, exchanging, and analyzing threat intelligence is not prevalent throughout the participating organizations yet. The feedback on Q2 shows that a visual representation of threat intelligence is highly preferred over a text-based representation. From the answers to our third question about the usability of the proposed tool, we can conclude that the tool is indeed usable. However, we received some suggestions for improvement. Especially the analysts who answered Q3 with a score of 3 or lower, provided helpful feedback. For instance, one comment recommended that nodes should not bump back to their original position after dragging to adjust the layout of the node-link diagram permanently. This and further received feedback was implemented into the subsequent version of KAVAS after this survey and before the expert interviews. Feedback to the tool's suitability and adequacy with respect to editing threat intelligence information (Q4) is very positive, as well. Moreover, the feedback to Question Q5 shows that KAVAS improves the understanding of incidents within the target group.

Expert interviews

In order to get in-depth insight into the support of the knowledge-assisted concepts in KAVAS, we implemented the suggestions for improvement from the survey and used the revised prototype for interviews with security experts to conduct a more detailed evaluation. The main goal of these interviews is to validate that KAVAS helps security experts to understand threat intelligence and that existing information can be enriched with expert domain knowledge. By showing the fulfillment of our prototype in terms of these two requirements, we can confirm that KAVAS indeed facilitates the internalization and externalization knowledge conversion processes. The remaining knowledge conversion processes, combination and collaboration, both are implicitly implemented in KAVAS: Threat intelligence can be inserted into the CTI Vault at any time through an API (*Combination*). Additionally, experts can collaboratively gain knowledge through externalizing their knowledge and making it accessible for other users (*Collaboration*). Therefore, our interviews focus on the internalization and externalization knowledge conversion process.

Participants

The interviewees are represented by five security experts from different sectors. We conducted interviews with a Chief Information Security Officer and a security analyst of an international machine manufacturer, with a Chief Technology Officer of a SME operating in the area of secure cloud services, with a consultant from a security consultancy as well as with an academic researcher in the field of IT security. None of the experts participated in the previous survey. Each participant has a medium to high knowledge regarding threat intelligence, while three of them deal with threat intelligence and related structured formats like STIX on a daily basis. However, none of the interviewed experts obtains a visual representation to facilitate this work.

Design & Procedure

The interviews with the experts are designed to follow a semi-structured approach according to Lazar et al.

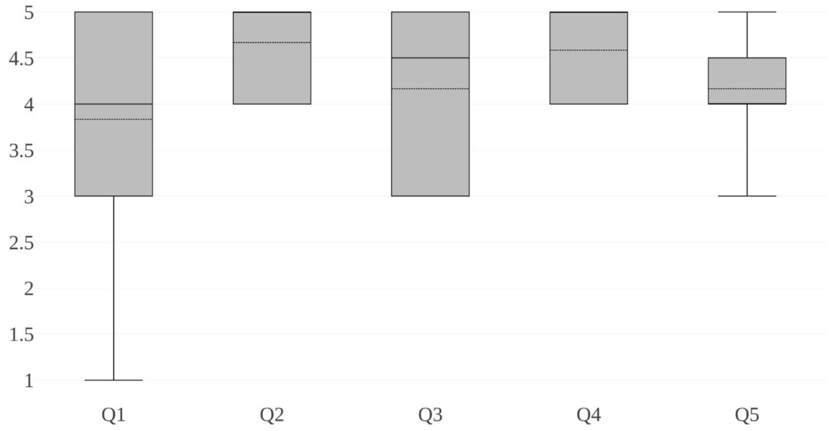


Fig. 6 Box plot reporting the KAVAS evaluation results. Solid lines mark the mean while dashed lines highlight the median

(Lazar et al. 2010). The interviews are separated into the following four phases:

- Phase 1) Introduction: At the beginning, every participant is questioned about their experience, such as their knowledge on CTI in general and on STIX. Afterwards, each expert receives a brief introduction into the STIX format and its problem of readability and accessibility. Thereby, the experts are asked to criticize any potential issues throughout the following interview phases. Next, each interviewee is guided to our prototypical web application. During the whole interview, the screen, of the participant using the tool, is shared with the interviewers.
- Phase 2) Internalization: To be able to test the intuitivity of the explorative analysis capabilities of KAVAS, the different interactive functionalities are not introduced in detail. The participants are asked to open a synthetic, previously designed STIX bundle (7 nodes, 8 links)[10] and to try to understand what happened in this bundle using the visual representation. In this phase, we pay special attention to the usage of interactions as well as to how the expert try to gain insight. After this first contact with KAVAS, the focus of the interview switches to a much more extensive bundle (65 nodes, 90 links).[11] With this bundle, we aim to discuss the scalability of the visual display in terms of the layout algorithm and the available interactions to adjust the layout. To conclude this phase of the interviews we ask for the experts' opinion on the tool so far and whether it supported them in understanding the threat intelligence information.
- Phase 3) Externalization: The focus of this phase is to test KAVAS' suitability to facilitate the externalization of domain knowledge, or more specifically, the insertion of new information and the modification of existing intelligence. To validate this with the interviewees, we provide a number of additional pieces of information and ask them to add this information to the previously explored smaller bundle. Again, we request them to give us feedback and criticize the tool whenever they have problems in understanding how it is working.
- Phase 4) Wrap-Up: The last phase of the interviews is dedicated to a summarizing discussion. Here, we discuss with the participant whether a more advanced version of KAVAS would be applicable to operative deployment and the conditions thereto. Finally, we collect a list of features and functionalities the interviewees find useful for improving the prototype.

Table 1 KAVAS survey results

	Q1	Q2	Q3	Q4	Q5
# Answers	12	12	12	12	12
Mean	3.83	4.67	4.17	4.58	4.17
Std dev	1.34	0.49	0.94	0.51	0.58
Min	1	4	3	4	3
Median	4	5	5	5	4
Max	5	5	5	5	5

Results

The interviews lasted between 45 to 70 min, which was mainly due to the summarizing discussion, where the experts brought up a lot of interesting points reaching from possible improvements of STIX itself to functionality features of KAVAS necessary for operative deployment in an organization. The results of the conducted interviews are presented in the following, divided according to the four phases described before.

- Phase 1) Introduction: At the beginning of each interview the participants are asked general questions to obtain basic data about the interviewees. Therefore, they are asked about their company as well as their exact role within the company. Furthermore, they are asked about their knowledge of Cyber Threat Intelligence and the STIX format in particular to determine their level of expertise. This first phase showed, that even though interviewees are familiar with threat intelligence information in general, they are rather unfamiliar with the specifics of the STIX format in most cases. Table 2 gives an overview on these general information about the interviewees.
- Phase 2) Internalization: Within this phase, the interviewees are asked to take a look at a predefined STIX bundle and to understand the contents of the presented incident. The interviews showed that KAVAS supports users to quickly understand an incident without having any previous knowledge. Especially the included filter functions of KAVAS turned out to be particularly helpful in this context. The consistently positive feedback within this phase showed, that the chosen representation is both suitable for representing incident information and makes it easily available for the user.

However, this phase also revealed some disadvantages and problems with the graph visualization in general and the realization in particular. While hassle-free usage was possible on large resolution displays, it turned out that problems arise when working on lower resolution displays, especially for handling larger datasets. The interviewees also missed some functionalities. For instance, they asked for advanced filter functions for different use-cases such as filtering the k-nearest neighbor nodes within specific tree sections. The interviews further revealed that existing filters and possible interactions with the user interface to re-structure the layout prove themselves as very useful features. It was also shown that the interface could be improved by implementing some additional features, such as on-demand windows displaying further information for objects with their associated relationships and an improved initial structuring of the presented graph representation. Altogether, the interviews show that KAVAS has a high utility for security specialists to convey and understand incident information. This manifested both in the assessment of the approach in general and the usability of the tool itself. However, it was also stated that a special training for employees might be necessary to cope with the complexity of STIX data. The interviewees also considered the tool to probably be helpful for practical usage. In this context they could for example think of a feed service to obtain incident information from a central authority, which could be used to understand attacks and prevent them from happening.

- Phase 3) Externalization: Within this phase, the interviewees are asked to use KAVAS to enrich the incident representation with additional, predefined knowledge made available by the interviewers. The process of editing information overall turned out to be mostly intuitive and easy to use for the experts.

Adding and editing nodes was perceived as intuitive by all participants, whereas some participants argued that editing relationships was a bit counter-intuitive when working with the tool for the first time. The fact that KAVAS distinguishes between explore and edit mode was perceived differently by the participants. While some accentuated the benefits of this clear separation, others found it cumbersome. However, the tool could be helpful to collect and enrich forensic evidence in e.g. CERT or incident response teams reconstructing how an incident compromised an organization. In this context, it was envisioned that this tool could especially be helpful within team meetings to collaboratively collect and edit threat intelligence information. It was also accentuated that there is most likely a need for integrity-proof intelligence data in the foreseeable future. Altogether, the enrichment of intelligence data was overall easy to use for the participants and mostly intuitive. The interview reveals that editing intelligence information is equally important to analyzing it. Moreover, the interviewees highlighted that there is an actual need for this feature within companies.

- Phase 4) Wrap-Up: Within the last phase, possible scenarios and conditions for an operative deployment of KAVAS and possible improvements for the prototype were discussed.

Table 2 General information on the interview participant

	Position	Business Branch	Organization's size	CTI Knowledge	STIX Knowledge
#1	Security Researcher	Academia	ca. 5.000	high	medium
#2	Chief Information Security Officer	Manufacturing	ca. 15.000	high	high
#3	Security Analyst	Manufacturing	ca. 15.000	medium	low
#4	Chief Technology Officer	Secure Cloud Services	ca. 60	medium	medium
#5	Senior Consultant	Security Consultancy	ca. 20	low	low

> One key problem revealed by the interviews is the question how threat intelligence data can be acquired. This concerns both the acquisition from external sources and the question how threat intelligence data can be produced within the company. In this context, it was also argued that there is a need for an automated generation of basic intelligence data that can be enriched by experts using tools like KAVAS afterwards. Integrating external intelligence feeds, cooperatively analyzing threat data as well as creating visual threat reports seems to be beneficial for companies. The interviewees also suggested several additional features to improve the user interface. These, for example, include improved highlighting for important and editable attributes or additional filter functions. Furthermore, the interviewees named some additional object properties that were necessary for practical usage, such as additional timestamps defining the point in time when the object was detected. These are not defined within the current STIX standard and consequently not available in KAVAS.

Discussion

The results of the conducted interviews show that KAVAS provides the ability for internalization and externalization of threat intelligence information. Given the fact, that it is still in the stage of a proof of concept prototype, the experts' feedback was already good. Furthermore, the experts provided several suggestions for future improvements of the tool.

The interviews also demonstrated that there is a strong interest for visualizing threat intelligence information among companies. The experts already have several use-cases for this kind of application in mind. However, the question of how to generate intelligence data in the first place remains.

Moreover, the interviews also showed that there are several weaknesses in the STIX standard, which became obvious while evaluating KAVAS. An example for this is the absence of a top-level element to represent and structure specific company assets such as IT systems affected by an incident.

Conclusion and future work

Conclusion

In this work we presented KAVAS, a concept for interactive visual analytics of threat intelligence information. Our approach persists information in a graph database to maintain an integrity-preserving data structure. This database is connected to a visual interface supporting security experts in understanding and analyzing incident descriptions. Additionally, the visual analytics component of KAVAS facilitates the process of including the knowledge of the security experts into CTI information. KAVAS achieves this with its functionalities to edit existing descriptions and adding new knowledge allowing for more thorough incident documentations.

While designing KAVAS, and especially its visual component, we aimed to follow the concept of knowledge-assisted visual analytics. More precisely we designed our concept to support the four main knowledge conversion processes which are essential to improve the collaboration of human and machines. *Internalization* is done in KAVAS by visually representing the incident documentations stored in the CTI vault. This way, the explicit knowledge in the CTI vault is accessible for security experts and they can gain knowledge using the visualization. KAVAS also supports *Externalization* as it allows for editing the STIX bundles. The tacit knowledge is externalized when the expert edits the threat intelligence information visually displayed in KAVAS. Being implemented as graph database the CTI vault has the essential functionalities to support the *Combination* knowledge conversion. This process is implemented in KAVAS as the CTI vault can be fed with new threat intelligence information and it includes this newly available knowledge into the existing knowledge base. A similar process in KAVAS supports the *Collaboration*. As externalization of an expert's tacit knowledge is possible, other security experts can profit from the externalized knowledge of each other providing an implicit form of collaboration.

The application KAVAS described throughout this work, clearly fulfills the three requirements we started with:

- **R1 - Handling of complex threat intelligence data:** The CTI Vault persists STIXbased threat intelligence information in a graph database. It additionally provides the possibilities to store externalized user knowledge in its knowledge base, while the integrity of the original information is preserved and ensured. Moreover, any data stored in the vault is compliant with the STIX format at any point in time.

- **R2 - Visual representation of STIX:** KAVAS' visual component can display threat intelligence and enables security experts to interactively explore incidents and gain insight about what happened.
- **R3 - Conversion of experts' knowledge**: As described above, KAVAS provides functionalities for each of the four knowledge conversion processes.

Fulfilling all the stated requirements, KAVAS offers a flexible platform for sharing, analyzing, annotating and visualizing cyber threat intelligence information based on the STIX data format.

Future work

Although we met the previously defined requirements for KAVAS, some challenges remain, which have to be addressed in future work.

A key challenge for future work regarding the CTI Vault will be the analysis of STIX data to find interconnections and redundancies between different bundles, which currently are standalone object pools, not attached to each other. Enabling the interconnections between and the merging of bundles could contribute greatly to the usage of STIX features. Additionally, this would improve the quality of available threat intelligence information. Examples for this are the merging of different incidents into a whole campaign of attacks and the determination of correlations between observed events within different incidents. The process for merging bundles and finding redundancies has to be subject for further research as it is a challenging task to identify interconnections and quality problems across independent bundles.

Additionally, there are some potential improvements regarding the functionalities of the visual component. During the interviews, the participants highlighted the need for a number of different advanced filters as well as some other features, which would help them even more to work with complex threat intelligence. Furthermore, experts should be included into the process of merging and connecting bundles. KAVAS could also be extended to support more sophisticated collaboration features for security experts like annotating CTI information to exchange domain knowledge in a more direct manner.

Another important future challenge regarding our proposed visual analytics tool is a comprehensive user study to quantify its effects on the work of security experts. These effects need to be quantified. Also the tool's impact on the quality of threat intelligence documentation has to be measured as expert knowledge can be externalized with KAVAS. Currently, KAVAS is only validated in terms of being able to work with the very limited examples provided by the OASIS committee and by a qualitative evaluation to show its feasibility. The main reason for this small-scaled evaluation is the lack of available real-world threat intelligence data being documented with STIX 2 up to this point in time. Its predecessor STIX 1 is the industry-wide state-of-the-art for documenting this type of information and we presume that it is very likely for STIX 2 to achieve the same amount of acceptance in the near future. Since the specification of STIX 2 is still under development, it is not reasonable to evaluate the effectiveness and efficiency of KAVAS in a comprehensive and quantitative manner yet.

Another topic for future work has to be the analysis and assurance of data quality among STIX bundles. As STIX supports collaborative efforts to maximize the number of prevented cyberattacks, the data quality of the incident descriptions is crucial. This is becoming even more true when the information is analyzed and enriched by human operators. High quality information is essential to ensure trust. Therefore, existing data quality metrics have to be applied on STIX-based descriptions to assess the added value they provide. Moreover, visual metaphors for these metrics have to be added to the KAVAS visual representation helping analysts to assess the trustworthiness of the information.

Endnotes

[1] https://trac.tools.ietf.org/html/rfc7970
[2] http://veriscommunity.net
[3] https://stixproject.github.io
[4] https://github.com/oasis-open/cti-stix-visualization
[5] https://www.oracle.com/technetwork/java/javaee/overview
[6] https://angular.io
[7] https://material.angular.io
[8] https://d3js.org
[9] https://oasis-open.github.io/cti-documentation/example/defining-campaign-ta-is/
[10] http://bit.ly/2NLDn3W
[11] http://bit.ly/2xX74EO

Acknowledgements
This research was supported by the Federal Ministry of Education and Research, Germany, as part of the BMBF DINGfest project (https://dingfest.ur.de).

Funding
Not applicable.

Availability of data and materials
Source code - CTI Vault

- Project name: CTI Vault
- Project home page: http://bit.ly/2LKFcgT

- Archived version: 1.0-SNAPSHOT
- Operating system(s): Platform independent
- Programming language: JavaEE
- Other requirements: Glassfish Application Server 4.1.1, JavaEE 6 or higher
- License: GNU GPL v3

Source code - Visual analytics component

- Project name: Visual analytics component
- Project home page: http://bit.ly/2LVn6YM
- Archived version: 1.1.0
- Operating system(s): Platform independent
- Programming language: HTML, Typescript
- Other requirements: Apache Webserver or similar, NPM 6.2.0 or higher
- License: GNU GPL v3

Authors' contributions

FM carried out the design and implementation of the CTI Vault. FB carried out the design and implementation of the visual analytics component. FM and FB conducted the evaluation and drafted the manuscript to equal parts. GP participated in the design of the different components and the study. GP also helped to draft the manuscript revising it critically for important intellectual content. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

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

Received: 6 August 2018 Accepted: 6 December 2018
Published online: 28 December 2018

References

Ackoff RL (1989) From data to wisdom. Journal of applied systems analysis 16(1): 3–9

Asgarli E, Burger E (2016) Semantic ontologies for cyber threat sharing standards. In: IEEE Symposium on Technologies for Homeland Security (HST)

Card SK, Mackinlay JD, Shneiderman B (eds) (1999) Readings in information visualization: using vision to think. Morgan Kaufmann, Burlington

Chang R, Ziemkiewicz C, Green TM, Ribarsky W (2009) Defining insight for visual analytics. IEEE Comput Graph Appl 29(2):14–17

Chen C (2005) Top 10 unsolved information visualization problems. IEEE Comput Graph Appl 25(4):12–16

Chen M, Ebert D, Hagen H, Laramee RS, van Liere R, Ma K, Ribarsky W, Scheuermann G, Silver D (2009) Data, information, and knowledge in visualization. IEEE Comput Graph Appl 29(1):12–19

Coleman J, Goettsch A, Savchenko A, Kollmann H, Wang K, Klement E, Bono P (1996) Teleinvivo™: towards collaborative volume visualization environments. Computers & Graphics 20(6):801–811

Endert A, Fiaux P, North C (2012) Semantic interaction for visual text analytics. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, New York

Fayyad U, Grinstein GG, Wierse A (2002) Information visualization in data mining and knowledge discovery. Morgan Kaufmann, Burlington

Federico P, Wagner M, Rind A, Amor-Amoròs A, Miksch S, Aigner W (2017) The role of explicit knowledge: A conceptual model of knowledge-assisted visual analytics. In: Proceedings of IEEE Conference on Visual Analytics Science and Technology (VAST). IEEE Computer Society Press, Los Alamitos

Heer J, Bostock M, Ogievetsky V (2010) A tour through the visualization zoo. Communications of the ACM 53(5):59–67

Heer J, Shneiderman B (2012) Interactive dynamics for visual analysis. Queue - Microprocessors 10(2):30

Keim D, Andrienko G, Fekete J-D, Görg C, Kohlhammer J, Melan.con G (2008) Visual analytics: definition, Process, and challenges. In: Information visualization. Lecture notes in computer science, vol 4950. Springer, Berlin, Heidelberg

Keim, D., Kohlhammer, J., Ellis, G., Mansmann, F. (eds.): Mastering the information age: solving problems with visual analytics, Goslar (2010)

Kobourov SG (2010) Force-directed drawing algorithms. In: Tamassia R (ed) Handbook of graph drawing and visualization. CRC Press, Boca Raton

Krasner GE, Pope ST (2000) A description of the model-view-controller user interface paradigm in the smalltalk-80 system. Journal of object oriented programming 1(3):26–49

Lazar J, Feng JH, Hochheiser H (2010) Research methods in human-computer interaction. Morgan Kaufmann, Burlington

Leichtnam L, Totel E, Prigent N, Mé L (2017) Starlord: Linked security data exploration in a 3d graph. In: IEEE Symposium on Visualization for Cyber Security (VizSec)

Luttgens JT, Pepe M, Mandia K (2014) Incident Response & Computer Forensics, 3rd edn. McGraw-Hill Education Group, Whitby

Marty R (2009) Applied security visualization. Addison-Wesley, Boston

Menges F, Pernul G (2018) A comparative analysis of incident reporting formats. Computers and Security 73:87–101

Nonaka I, Takeuchi H (1995) The knowledge-creating company: how Japanese companies create the Dynamcis of innovation. Oxford University Press, Oxford

Piazza R, Wunder J, Jordan B (2017a) STIX™ version 2.0. Part 1: STIX Core concepts. OASIS committee

Piazza R, Wunder J, Jordan B (2017b) STIX™ version 2.0. Part 2: STIX objects. OASIS committee

Pike WA, Stasko J, Chang R, O'Connell TA (2009) The science of interaction. Information Visualization 8(4):263–274

Pirolli P, Card S (2005) The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In: Proceedings of International Conference on Intelligence Analysis McLean, VA, USA

Polanyi M (1983) The tacit dimension. University of Chicago Press, Chicago

Sacha D, Stoffel A, Stoffel F, Kwon BC, Ellis G, Keim D (2014) Knowledge generation model for visual analytics. IEEE Trans Vis Comput Graph 20(12): 1604–1613

Sauerwein C, Sillaber CN, Mussmann A, Breu R (2017) Threat intelligence sharing platforms : An exploratory study of software vendors and research perspectives. In: 13. Internationale Tagung Wirtschaftsinformatik, WI 2017, St. Gallen

Severino, R.: The data visualisation Catalogue (2018). https://datavizcatalogue.com/index.html. Accessed 2018-08-03

Shackleford D (2015) Who's using Cyberthreat intelligence and how? SANS institute, Swansea

Shackleford D (2016) SANS 2016 Security Analytics Survey. SANS Institute, Swansea

Shneiderman B (1996) The eyes have it: A task by data type taxonomy for information visualizations. In: Proceedings of the 1996 IEEE Symposium on Visual Languages. IEEE Computer Society Press, Los Alamitos

Shrinivasan YB, van Wijk JJ (2008) Supporting the analytical reasoning process in information visualization. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, New York

Staheli D, Yu T, Crouser RJ, Damodaran S, Nam K, O'Gwynn D, McKenna S, Harrison L (2014) Visualization evaluation for cyber security. In: IEEE Symposium on Visualization for Cyber Security (VizSec). ACM, New York

Theron R, Magán-Carrión R, Camacho J, Fernandez GM (2017) Network-wide intrusion detection supported by multivariate analysis and interactive visualization. In: IEEE Symposium on Visualization for Cyber Security (VizSec). IEEE Computer Society Press, Los Alamitos

Thomas JJ, Cook KA (eds) (2005) Illuminating the Path: The Research and Development Agenda for Visual Analytics. IEEE Computer Society Press, Los Alamitos

Wagner M, Rind A, Thür N, Aigner W (2017) A knowledge-assisted visual malware analysis system: design, validation, and reflection of Kamas. Computers &Security 67:1–15

Wang X, Jeong DH, Dou W, Lee S-W, Ribarsky W, Chang R (2009) Defining and applying knowledge conversion processes to a visual analytics system. Computers & Graphics 33(5):616–623

Yen J, Erbacher RF, Zhong C, Liu P (2014) In: Kott A, Wang C, Erbacher RF (eds) Cognitive Process. Springer, Cham

Zhong C, Yen J, Liu P, Erbacher RF (2018) Learning from experts' experience: toward automated cyber security data triage. IEEE Systems Journal:1–12

4 Human-as-a-security-sensor for harvesting threat intelligence

Current status:	Published	
Journal:	Cybersecurity, Volume 2, Number 1, December 2019	
Date of acceptance:	29 August 2019	
Full citation:	Manfred Vielberth, Florian Menges and Günther Pernul. Human-as-a-security-sensor for harvesting threat intelligence. *Cybersecurity 2(1): 23 (2019).*	
Authors contributions:	Manfred Vielberth	45%
	Florian Menges	45%
	Günther Pernul	10%

Journal Description: The Cybersecurity journal aims to systematically cover all essential aspects of cybersecurity, with a focus on reporting on cyberspace security issues, the latest research results, and real-world deployment of security technologies.

Vielberth *et al. Cybersecurity* (2019) 2:23
https://doi.org/10.1186/s42400-019-0040-0

Cybersecurity

RESEARCH Open Access

Human-as-a-security-sensor for harvesting threat intelligence

Manfred Vielberth*, Florian Menges and Günther Pernul

Abstract
Humans are commonly seen as the weakest link in corporate information security. This led to a lot of effort being put into security training and awareness campaigns, which resulted in employees being less likely the target of successful attacks. Existing approaches, however, do not tap the full potential that can be gained through these campaigns. On the one hand, human perception offers an additional source of contextual information for detected incidents, on the other hand it serves as information source for incidents that may not be detectable by automated procedures. These approaches only allow a text-based reporting of basic incident information. A structured recording of human delivered information that also provides compatibility with existing SIEM systems is still missing. In this work, we propose an approach, which allows humans to systematically report perceived anomalies or incidents in a structured way. Our approach furthermore supports the integration of such reports into analytics systems. Thereby, we identify connecting points to SIEM systems, develop a taxonomy for structuring elements reportable by humans acting as a security sensor and develop a structured data format to record data delivered by humans. A prototypical human-as-a-security-sensor wizard applied to a real-world use-case shows our proof of concept.

Keywords: Cyber threat intelligence, Human awareness, Human-as-a-security-sensor, Security information and event management (SIEM)

1 Introduction

Today's security analytics solutions like Security Information and Event Management (SIEM) systems heavily rely on a huge amount of data in order to reliably detect incidents in organizations (Bhatt et al. 2014). New sources providing security-relevant data, such as knowledge about occurred incidents observed by human individuals, can therefore significantly enlarge the data basis for incident detection.

During past years, humans or employees were generally seen as the weakest link in corporate IT security (Lineberry 2007). To mitigate the risk of humans for IT security, a lot of effort is put into awareness campaigns and training of employees (Mello 2017) to ensure that they receive a basic understanding of this topic. This also enables them to distinguish between "normal" events and events harming the organization. However, the ability to recognize malicious events is not harnessed to its full extent. Information about potential incidents might be hidden in the minds of humans and could be the missing link for attack detection or for forensic reconstruction of adverse events. Especially when it comes to nontechnical traces. Therefore, we argue that the connection of digital events with non-digital events observed by people is crucial to IT security.

In this paper, we describe an approach that integrates the human data source to further processing in security analytics systems (e.g. SIEM systems). Therefore, we illustrate the problem with a motivating example in Section 2. Subsequently, related work in the area of human-as-a-security-sensor is portrayed within Section 3. In Section 4, we present the problem and research question tackled and show how to integrate human sensors into SIEM systems in Section 4.1. In Section 4.2, a risk model and a taxonomy for human threat reporting are proposed. On this basis we develop a CTI base data structure for human sensor information in section 4.3 and a data format for the representation of this data in Section 4.4. Finally, the proposed approach is evaluated in Section 5 and concluded in Section 6.

*Correspondence: manfred.vielberth@ur.de
Universität Regensburg, Universitätstr. 31, 93053 Regensburg, DE, Germany

2 Motivating example

In the following section, we use a real-world attack to illustrate the main problem tackled in this work. The example underlines benefits that may arise from integrating the human factor into threat detection mechanisms, including improved threat detection and additional context information.

Between 2017 and 2018, Kaspersky Lab (Golovanov 2018) investigated several cybersecurity incidents that go by the name of DarkVishnya. Malicious devices were directly connected to organizations' local networks, causing damage estimated to multiple millions of dollars. As shown in Fig. 1, the attack was conducted in the following essential steps:

1 The attacker tries to physically enter the premises of the attacked organization, claiming to be a person with legitimate interest (e.g. being an applicant or a courier).
2 After the successful entrance, the attacker tries to place a network device unobtrusively and hides it by blending it into the surrounding area. Moreover, the device is connected to the local network infrastructure in order to enable further attack steps.
3 After the attacker has left the organization, the placed device is remotely accessed by utilizing standard mobile technologies like GPRS, 3G or LTE to control it for further attack steps.
4 The attacker scans the network for usable information and for accessible resources in the local network. This may include shared folders, servers or other systems that execute critical actions. Additionally, brute-force attacks or network sniffing is used to gain access to login credentials.
5 The attacker tries to exploit the previously gained access e.g. by installing malware to retain access and to execute malicious services.

The crux of the attack is that the first three steps are nearly impossible to detect with technical security systems like SIEM, or Intrusion Detection Systems (IDS), as neither the attacker entering the building, nor the placing of a hardware device leave any digital traces. The first digital traces that may be detected by security systems are left at the beginning of the network access. Unlike automated analyses, employees have the ability to detect and report such anomalies before technical traces and potential damages occur. If, for example, a suspicious person walks around the office building, the employee might already categorize this event as an anomaly. Additionally, context information, such as a description of a person, enhances this first perception. However, employees are often not able to recognize technical traces, such as network scans. The example demonstrates that it is hardly possible to capture the full extent of an attack, when collecting technical or human traces independently or if one of them is not considered at all. Therefore, we propose an approach that enables the acquisition of anomalies or potential attacks detected by employees, to translate them into machine readable language and thus to create the basis for combining these two types of data.

3 Related work

The first IT security related approaches for threat reporting by humans are systems that handle malicious or unwanted emails. These can be narrowed down to spam and phishing emails. There are several examples available in practice that allow to report such threats. These are in most cases integrated into email software, where emails can be marked (Google LLC; Microsoft Corporation) or a standalone web interface is provided (Anti-Phishing Working Group). In most cases, these reports are used to train phishing or spam filters of the provider.

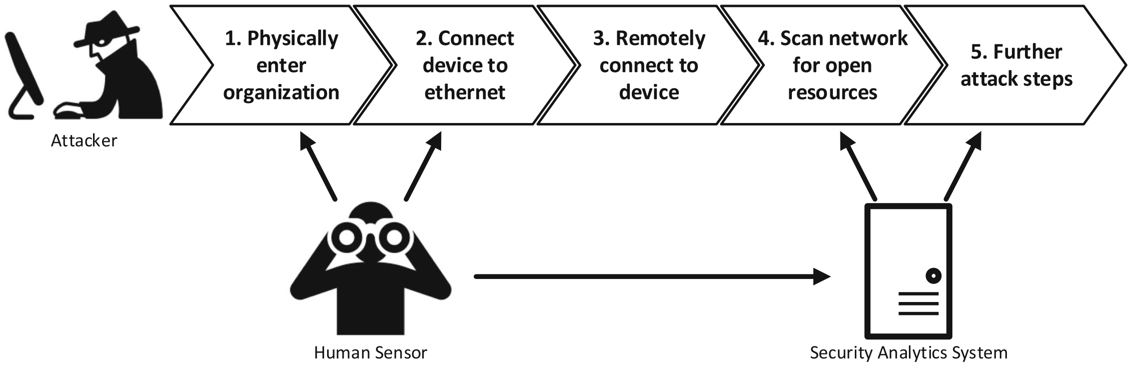

1. Physically enter organization
2. Connect device to ethernet
3. Remotely connect to device
4. Scan network for open resources
5. Further attack steps
Attacker
Human Sensor
Security Analytics System

Fig. 1 Attack steps of DarkVishnya

A second approach commonly applied in practice is human-to-human reporting. A central contact point (e.g. the help desk of an organization) is set up. Especially when implementing an information security management system (e.g. control A.13.1 of the ISO 27001 standard demands the reporting of security events or weaknesses from all employees (ISO/IEC 27001: Information technology – Security techniques – Information security management systems – Requirements 2013)) this is common practice for reporting security issues by employees (Hintzbergen et al. 2015). However, this approach entails some disadvantages. For example, the human point of contact has to interpret the received information and decide how to proceed. This might result in wrong decision-making, especially as help desk personnel are commonly no security specialists. Additionally, the collected data is poorly structured and not utilizable for technical analyses in most cases. Although not security related, the idea of using humans as sensors has been a topic of interest for a while. For example, Wang et al. (2014) pursue the idea that social networks might be the largest existing human sensor networks. Furthermore, Kostakos et al. (2017) investigate several scenarios, where humans can act as sensors. They consider, among others, crowdsourcing markets, social media and the collection of citizen opinions.

Heartfield and Loukas (2018) recently proposed a more general approach focused on semantic social engineering attacks. In their work, they develop and prototypically implement a framework for reporting semantic social engineering attacks. They propose a model for predicting the reliability of reports generated by humans and show, that human sensors can outperform technical security systems in their considered context. In addition, they implement a backend application, which is mainly responsible for incident response and dashboard capabilities. In one of their previous works (Heartfield et al. 2016), they also coined the term human-as-a-security-sensor , which refers to the ”paradigm of leveraging the ability of human users to act as sensors that can detect and report information security threats”. For our work, we adopt the meaning of the paradigm. This capability is strongly influenced by the security training the person received in advance. In addition, an approach for scoring the trustworthiness of human sensors was introduced by Rahman et al. (2017). They especially monitor features of the mobile device, utilized for conducting the report, for predicting the reliability of the provided data.

To sum up the developments in this area, platforms for reporting potential malicious or unwanted emails were implemented at first. This was followed by the development of processes for human-to-human reporting and succeeded by more sophisticated approaches for detecting semantic social engineering attacks with the help of a human-as-a-security-sensor framework. However, to the best of our knowledge, there are no approaches that support reporting a wide range of possible attacks detectable by humans. Additionally, there are no concepts for integrating reported incidents into existing, and in many organizations already established, security systems (e.g. SIEM systems). Moreover, the participation of people with different knowledge in the field of cybersecurity, is currently neglected.

4 Integrated human-as-a-Security-Sensor (IHaaSS)

Resulting from the explanations in Section 2 and Section 3 we tackle the issue, that **observations of humans are either poorly or not at all integrated into the automatic security analytics process**. This raises the following research questions:

Q1: What are the connection points of a human-as-a-sensor to the data flow of a SIEM system?

Q2: How can human-provided information be structured (data format) in order to facilitate further technical processing?

Q3: How can incident information be systematically acquired from people?

To answer these research questions, we applied the following approach:

1. To answer Q1, we illustrate how to integrate human-as-a-security-sensors into security analytics in Section 4.1. This is based on existing data collection approaches and the generic data flow of SIEM systems identified in literature (Vielberth and Pernul 2018).
2. To answer Q2 and Q3 it is in a first step necessary to identify all possibilities a human sensor can report. This is carried out by developing a risk model and taxonomy, adhering to the method for taxonomy development by Nickerson et al. (2013) in Section 4.2.
3. To answer research question Q2, we first conceptualize a CTI base data structure for the representation of human sensor data in Section 4.3. On this basis we then identify suitable CTI data format standards to realize this base data structure and extend them for the capturing of human sensor data in Section 4.4. This allows the integration of human-generated reports into SIEM systems for further processing.
4. Finally, the incident information can systematically be acquired (Q3) following the risk model and taxonomy, which is restricted by constraints identified in Section 4.5.

Thereby, we see the main contributions of this paper in the identification of connection points, the development of the taxonomy, the extension of well-established data formats and the identification of constraints for a systematic data acquisition. We also show the practicability of our approach using a prototypical implementation and an exemplary real-world use case.

4.1 Integrating human sensors into SIEM

To connect technical data with human-generated traces, both need to be brought together in one single system. One way to achieve this is to integrate human knowledge into SIEM systems that are already in place in most organizations within a security operations center (SOC) (Crowley and Pescatore 2018). Apart from that, the presented approach can easily be adapted to other security monitoring tools.

A SIEM system is essentially designed for the collection of relevant log data to detect incidents and gain situational security awareness. In Fig. 2 we extended the basic SIEM structure as proposed by Vielberth and Pernul (2018) with the data flow of an integrated human-as-a-security-sensor. Hereby, the SIEM first *collects* relevant event information, in most cases in the form of log data. This data gets *enriched* with additional context data and translated in a uniform representation during the *normalization* step. The core of the system lies in the *correlation and analysis* component, where information from various sources is connected and incidents are detected using methods such as pattern matching. *Monitoring* enables security analysts to be actively involved in the analysis, whereas *reporting* delivers compliance reports or enables the participation in established threat intelligence sharing platforms between organizations. In case of a detected incident, *alerting and incident response* triggers necessary reactions to mitigate further harm. Finally, the *storage* module is responsible for both, short- and long-term storage of event data and analysis results.

For integrating human sensors into SIEM (Q1), we extend the basic SIEM data collection approach. According to Holik et al. (2015) and Turnbull (2019), two fundamental approaches can be applied. They both distinguish between push- and pull- based log collection. Since we do not collect log data, but human-generated incident information, these two approaches require adaptation. In the following, both approaches are described in the context of this paper:

- **Push:** The push method applies when an employee initially detects an incident and actively delivers the gathered information to the system. It is important to offer guidance for enabling humans to provide information in a structured way, especially if their knowledge about security is limited. Additionally, employees might report information in different levels of detail, depending on how much they know about the incident. The push approach is similar to systems pushing log data into SIEM systems as described in literature (Holik et al. 2015). Thus, the connection point of the push approach is the *event collection* (compare Fig. 2).
- **Pull:** In traditional SIEM systems, the pull approach basically refers to polling-based systems (Turnbull 2019), which query the data periodically, generally in fixed time intervals. Since periodically polling information from human sensors is hardly feasible, we only pull information in certain cases. These cases occur during certain steps of the SIEM data flow (as described subsequently), which are the connection points for the pull approach. The pull approach is applied if important information is missing during the *monitoring* or *analysis* of incidents. Presumably, this happens in case an incident is reported by people with little knowledge about IT security or about the context of the incident. The lack of information can

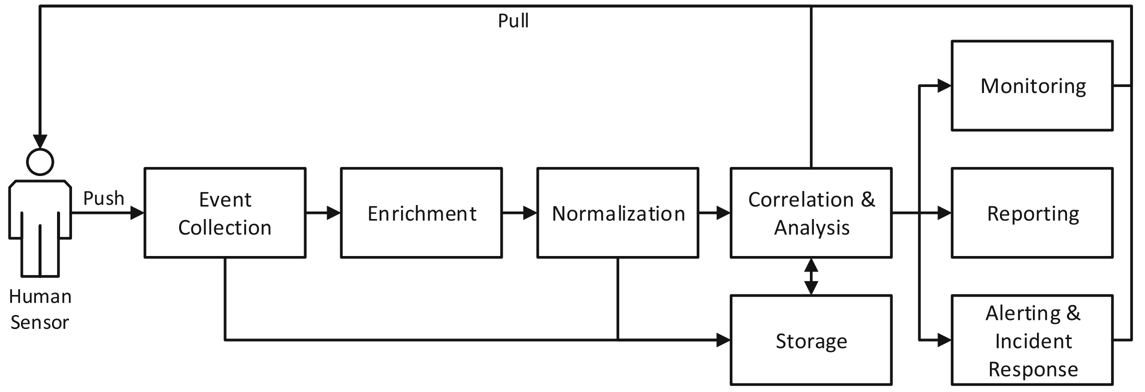


Fig. 2 Integrating human-as-a-security-sensor into the SIEM Data Flow

either be detected automatically during the *correlation and analysis* phase, or by human experts monitoring the system or during *incident response* steps. Furthermore, needed information might be missing in case technical indications about an incident occur, but previous attack steps were not reported. For instance, technical traces from step four of the attack in Fig. 1 could be identified in the system, while previous attack steps were not reported. Therefore, it is necessary to advice employees to report missing hints. In order to gain more information, an expert can interview the reporting person and guide him to contribute further or more detailed information.

4.2 IHaaSS incident model and taxonomy

For being able to develop a format (Q2) and structure the acquisition of information (Q3), it is necessary to capture everything that human-security-sensors can perceive. Information security management standards and its associated resources provide a good basis by providing risk assessment guidelines. These consider and evaluate mostly future risks. However, in our approach we want to report past incidents, requiring to adjustment for some elements. Regarding the NIST Guide for Conducting Risk Assessments (Joint Task Force Transformation Initiative 2012) and Juliadotter and Choo (2015), the key risk factors are *Threat Sources*, *Threat Events*/*Vector*, *Targets*/*Vulnerabilities* and *Impact*. All four risk factors are observable or can at least be assessed by human sensors and thus, have to be dealt with. The resulting threat model can be seen in Fig. 3.

In the proposed Integrated Human-as-a-Security-Sensor (IHaaSS) incident model, there are two types of threat events: threat events caused by humans or technical sources (commonly security events) and events which are not necessarily assignable to a source (especially safety events). Threat events can be either initiated by threat sources or by previous events. Furthermore, it is possible that no entities are affected, or the affected entities are not (yet) known. The same applies to the expected impacts. This leads to the conclusion that only threat events are mandatory elements, as without threat events there is no need to report.

In order to get a deeper insight into human sensor reports, we examine the four risk factors in more detail, thereby create a taxonomy for human-as-a-security-sensor threat reporting. This taxonomy classifies and

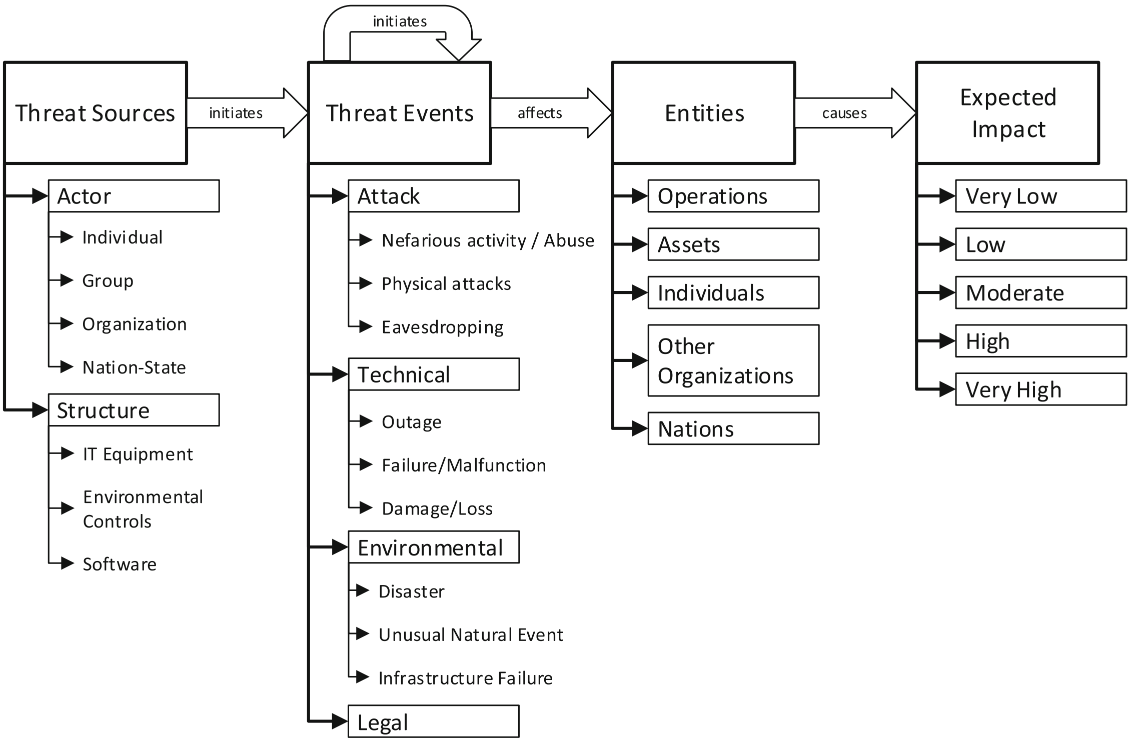


Fig. 3 IHaaSS Incident Model and Taxonomy

structures the security-related artifacts a human sensor can observe. Thereby, we loosely adhere to the method for taxonomy development by Nickerson et al. (2013). However, we did not develop a completely new taxonomy, but rather combined and adapted existing taxonomies to fit the purpose. Thereby we followed the "conceptual-to-empirical approach" (Nickerson et al. 2013), because of the existing foundations and a well-established knowledge base in this area. The identified objects are described in more detail in the following:

- **Threat Sources:** Threat sources are the starting point of the incident and can initiate subsequent threat events. This part of the taxonomy is based on the NIST Taxonomy of Threat Sources (Joint Task Force Transformation Initiative 2012). However, to avoid overlaps with subcategories of threat events, we narrow the scope. In the context of our paper, a threat source is an entity, which can decide and initiates events. Thus, the environment defined as a threat source by NIST is equal to a threat event in our taxonomy as we argue that environmental factors cannot take decisions. Additionally, environmental events might be initiated by sources and are therefore better classified as events (e.g. a fire can be set by a person). However, the risk model process remains unaffected, because events can initiate other events. As a result, environmental events can still trigger structural events such as outages.
- **Threat Events:** Threat Events are the processes actually causing harm to an organization and thus are the key component of an IHaaSS report. Our taxonomy of threat events is based on the ENISA Threat Taxonomy (Marinos 2016) with some changes in order to fit in the rest of our model. We have defined more general categories, which allow the distinction between intentional (Attack) and potentially unintentional (Technical, Environmental, Legal) events. This is especially important in order to form dependencies in Section 4.5. Furthermore, the environmental events are merged with environmental threat sources from the NIST Taxonomy of Threat Sources (Joint Task Force Transformation Initiative 2012).
- **Entities:** The identification of relevant assets (asset inventory) is much discussed in academic literature and by industry, due to its importance to risk management (Fenz et al. 2014). Our approach, however, is somewhat broader, which is why we talk about entities (e.g. other organizations may be affected, which are not necessarily an asset for the company). The entity taxonomy is taken from (Joint Task Force Transformation Initiative 2012), wherein it is called adverse impact.
- **Expected Impact:** The expectation of possible impacts is usually quite hard to classify for humans. Therefore, the human sensor commonly provides qualitative estimations, especially when the IT security knowledge is low. Nevertheless, this estimation can be very helpful for evaluating further actions and reactions. Very Low to Very High is a rating of the effect of the event as described by the NIST (Joint Task Force Transformation Initiative 2012). It ranges from "negligible" to "multiple severe or catastrophic effects".

4.3 Conceptualizing a CTI data structure for human sensor data

In the previous sections, we introduced connecting points for the integration of human knowledge into SIEM systems and developed a taxonomy that serves as an information basis for the acquisition of threats detected by humans. In this section, we lay the theoretical foundations for the integration of human-provided information into SIEM data processing. The central factor for this integration is the harmonization of data structures to ensure compatibility of information. As shown in 4.1, SIEM systems work with both normalized raw data and enriched context data, which can be summarized under the term Cyber Threat Intelligence (CTI). To enable the integration of these types of information, we propose an approach of translating the human provided information into the existing CTI data structures in this section. To this end, we first discuss the types of information that can be provided by human sensors and classify them in the context of CTI information. On this basis, we then propose a CTI data structure that allows to fully capture information provided by human sensors to answer the research question Q2 on a general level. Finally, Table 1 summarizes the results of this section

The work of Burger et al. (2014) serves as a basis for the allocation of human sensor information to CTI data structures. It divides CTI into the three main categories *Intelligence*, *Attribution* and *Indicator*. *Intelligence* refers to rather complex issues such as concrete procedures of attackers or methods for mitigating security incidents, which cannot be fully acquired from automated analyses. Although a deeper expert knowledge is necessary for the final evaluation of intelligence information, untrained employees can contribute valuable information, which may make an incident detection possible in the first place. An example of this would be the detection of unauthorized physical access to protected resources. The *Attribution* category describes various types of additional contextual information about a security incident. These include, for example, information on attackers or affected devices.

Table 1 CTI base model extensions

Classification	Taxonomy		UPSIDE Model	Changes
Intelligence	Threat Events	Attack	Attack Event	-
		Technical	-	Technical Event
		Environmental	-	Environmental Event
		Legal	-	Legal Event
	Expected Impact		Result	Result
Attribution	Threat Sources	Actor	Attacker	Actor
		Structure	-	Structural Source
	Entities	Assets	Attack Target	Affected Entity
		Persons	Attack Target	Affected Entity
Indicator	Threat Events		Indicator	-

This data is also only recognizable to a limited extent through automated analyses. Since attribution information usually does not require specific specialist knowledge, employees can also make a valuable contribution here. For example, employees can help identifying a potential attacker and point out potentially affected devices. In contrast to these categories, *Indicator* describes specific system events that can, for example, be obtained from system logs. Since log files contain extensive information, they are usually evaluated using automated analyses and can only be used to a limited extent within a human sensor platform. However, when an incident is captured, additional fine-granular information may also be provided. For example, a malicious email provides information about a potential attack or an attacker, but also provides fine-grained information within its source code. As a result, indicator information is not primary information that is obtained from human observations, but secondary information that is collected when entities are created and populated. Summarizing, it can be stated that human sensors can mainly contribute to analyses with context information from the categories *Intelligence* and *Attribution* whereas *Indicator* information is only used to a very limited extent.

After performing a classification of human sensor data in the context of CTI data structures, we propose a CTI data structure that is able to cover the full range of human sensor information in the following. To achieve this, we utilize the previously introduced categories *Intelligence*, *Attribution* and *Indicator* to describe the individual changes necessary. More specifically, we use the UPSIDE model that describes CTI base entities by Menges and Pernul (2018) to determine and discuss entities that can be mapped by CTI data structures and those that are still missing for the representation of human provided information. On this basis, we propose conceptual adaptations to existing CTI data structures to support human sensor data as described in our taxonomy.

- **Intelligence:** The *Intelligence* category describes the attack patterns used, countermeasures taken and additional information on incidents such as the expected impact. The Threat Events and Expected Impact sections of the taxonomy can be assigned to this category. Threat events are divided into active (attack) and passive (technical, environmental and legal) incidents. According to the CTI base model, the description of active attacks is possible by defining attack events and the underlying procedure. Incidents without an active component are not supported so far. In addition, the model offers the possibility to define the result of an attack as result entity. This allows "Expected Impact" from our taxonomy to be mapped, however, this also only applies for active attacks.
- **Attribution:** The *Attribution* category defines various contextual information, such as information about attackers and targets. The sections Threat Sources and Entities from the taxonomy can both be assigned to this category. In the area of threat sources, the CTI base model can represent active attackers. Although, an unintentionally involved actor and other threat sources cannot be defined yet. In the taxonomy section entities, both assets and persons can be represented within the CTI base model. However, these can only be represented as targets in connection with an attack. It is not possible to represent any other kind of participation of these entities.
- **Indicator:** The *Indicator* category is used to display detailed information within threat events. The entity indicator from the CTI base model defines a generic representation within a security incident that can be assigned to any other entity. Accordingly, the requirements of the taxonomy are basically fulfilled in this area.

After comparing our taxonomy with the capabilities of the CTI base model, we discuss necessary adjustments for the integration of human sensor information in the following. Several adjustments are necessary within the *Intelligence* section. Since only attack events are supported, it is necessary to introduce additional entities to be able to map passive events. This includes technical events, environmental events and legal events. In addition, the result of an event must be adapted in such a way that the result of passive events can also be represented. The *Attribution* area also requires several adjustments. On the one hand, the attacker element must be extended in such a way that a passive participant can also be represented. In addition, it is also necessary to introduce an additional entity to represent a structural source for incidents. Finally, entities can be represented completely, but only in the context of an attack. Here an appropriate extension is necessary so that entities can also be affected by passive events. The indicator area does not require any adjustments at the conceptual level. Summarizing, Table 1 gives an overview of the results of this section. Column Classification assigns the results to the respective CTI category, while column Taxonomy shows the elements of the taxonomy under consideration. The UPSIDE Model column shows the assignment to the CTI base model and column Changes shows the necessary adjustments to the base model to support human sensor information.

4.4 A structured representation for threat intelligence reported by humans

In the previous sections, we introduced connection points for integrating human knowledge into SIEM systems and a taxonomy that defines the information basis for the acquisition of threat information detected by humans. Subsequently, we introduced the theoretical foundation for a CTI data structure that is able to represent human sensor data. Based on these findings, we develop a CTI data format in this section that allows to capture information provided by human sensors and enables further technical processing according to research question Q2.

In developing the data format we pursue two main objectives. On the one hand, we aim to achieve a high compatibility to existing SIEM systems to allow a direct integration of additional information into the system. On the other hand, we aim to create a format that allows a complete representation of human sensor data. More specifically, the full scope of the taxonomy shown in Section 4.2 needs to be covered. In order to meet these requirements as completely as possible, we first select existing and well supported CTI data format standards as development basis in the following. Subsequently, we propose a specification of necessary extensions for the integration of human sensor data according to Section 4.3.

Event collection modules within SIEM systems handle heterogeneous raw data from different log sources. This data is then translated into homogeneous indicator data structures to allow further processing. Literature provides different standards for the structured representation of indicators, such as CybOX[1] or openIoC[2]. These data structures are commonly referred to as Indicators of Compromise (IoC) as they depict a set of observations associated with a threat (Appala et al. 2015). These basic incident data can furthermore be enriched using intelligence- and attribution data, such as information about attackers, utilized attack patterns or attackers' objectives as shown by Burger et al. (2014). Together, they allow the representation of complex security incident information as shown in Section 4.3. Literature also offers different standards for representing enriched incidents information, such as STIX, IODEF, VERIS and X-ARF (Barnum 2014; Dandurand et al. 2015; Menges and Pernul 2018). In order to allow the representation of human delivered information, we chose the combination of the existing formats CybOX and STIX as development basis. Both formats are issued together by MITRE[3] and a combined usage is explicitly intended. Since these formats are most commonly applied to represent comprehensive threat intelligence information (Shackleford and SANS Institute 2015; Sauerwein et al. 2017), high compatibility to existing systems can be assumed. Moreover, they offer broader representation capabilities in their basic configuration than comparable formats as shown by Menges and Pernul (2018) and therefore, represent a solid foundation for the integration of human delivered information. Both CybOX and STIX are briefly introduced in the following and examined for necessary extensions to represent human delivered information afterwards.

CybOX provides an extensive catalog of object types for the description of the indicator layer. Each object represents individual components of log files, such as files, processes or network packets and offers description options at a detailed level. For example, the object type *file* allows the description of basic file properties such as path, extension or file name but also additional information such as permissions, compression procedures or creation date. STIX is the most extensive and widespread format for the structured representation of cyber threat intelligence information available today (Burger et al. 2014). It provides flexible data structures, such as non-structured free-text attributes, built-in controlled vocabularies using predefined values (vocabs) as well as integrated references to external data sources such as platform or vulnerability databases (enumerations). STIX uses indicators provided by CybOX as information basis and a wide range of well-defined data definitions to express the intelligence and attribution information for threats. The data model consists of the following core concepts. Incident is the central

[1] https://cyboxproject.github.io
[2] https://github.com/mandiant/OpenIOC_1.1
[3] https://www.mitre.org

entity for structuring the incident information. TTP (Tactics, Techniques and Procedures) and Course of Action to describe the Intelligence layer. Campaign, Threat Actor and Exploit Target describe the Attribution layer. Indicator, Observable serves as interface to the Indication layer that is essentially provided by CybOX. Moreover, numerous attributes for a detailed expression of these concepts are provided by the data model (Barnum 2014; Menges and Pernul 2018; Fransen et al. 2015).
After this short introduction of the data formats STIX and CybOX, we develop adjustments for these formats to represent human delivered incident information following the IHaaSS taxonomy (see Section 4.2) and CTI data structure (see Section 4.3) in the following. For this purpose, we first discuss the missing elements within the data formats based on the CTI basic data structure. On this basis, we propose the following changes to the formats to allow the integration of human sensors.

- **Intelligence:** Previously, it was shown that attack events can be mapped within the CTI base model, whereas other events are not available yet. Using the taxonomy, we are able to limit these additional events to the categories *Structural*, *Environmental* and *Legal*. In order to also support these events within the data format, we have defined the additional entities "Technical Event", "Environmental Event" and "Legal Event". All these entities are derived from the basic entity TTP, which describes tactics, techniques and procedures used in the course of an attack. An essential property of TTP objects is the structured representation of attack patterns. For this purpose, STIX uses the CAPEC Enumeration, a freely available data set of known attack patterns for the unambiguous description of specific attacks. In order to achieve a comparable functionality for the additionally defined events, we defined a corresponding vocabularies for structural, legal and environmental events. Each vocabulary offers predefined event definitions according to our taxonomy. In addition to the event definitions, the area of intelligence also offer possibilities for describing the expected impact of an incident. For this purpose it was previously shown that the base model only provides impact definitions that emerge from active attacks. Although this is basically also true for the data format, its data definitions do not explicitly restrict the representation of incident results to an underlying attack. As a result, no changes are necessary to enable the definition of specific event results.
- **Attribution:** It was shown that an integration of structural sources is necessary for addressing passive threats within the CTI base format. In addition, it was shown that the entities are limited to the expression of active attacks. The data format already provides elements such as Threat Actor, Exploit Target to represent active attacks and attackers, and Asset Vocabulary to define assets. To enable the integration of passive threats, we extend STIX with the definition of an additional entity "structural source" as intended in the CTI base format. Since this is an alternative threat source, the object is derived from the existing Threat Actor object and exists on the same level. This object is extended by an additional vocabulary "StructuralSourceTypeVocab" to be able to represent structural threat sources in a structured way. Since this extension of threat sources also extends the scope of attribution, we additionally defined an extension of the asset vocabulary. This makes it possible to define additional assets that can occur in connection with passive threats.
- **Indicator:** The indicator category is used to represent incident event information on a high level of detail, which are basically able cover the event information that may be delivered by humans. However, humans are usually not capable of delivering information on this level of detail and will rather provide unstructured data fragments. Consequently, such data fragments must be evaluated afterwards and the format must allow the unstructured data to be recorded at the time of acquisition. For this reason, we have also added an extension to the Observable object that allows to include unstructured data, which can later be translated into structured CybOX information.

In addition to these specific extensions, all objects were equipped with specific IHaaSS IDs and to enable additional references between the objects. This allows employees to express their perception by establishing links between objects. These additional connections can then be separately evaluated by analysts and integrated into the analysis results. In summary, it was shown in this section that STIX already fulfills numerous requirements for the implementation of an IHaaSS platform. However, the format requires different extensions to fully match the taxonomy according to the CTI base model. To achieve this, additional entities to represent structural threat sources as well as environmental, structural and legal events are defined within the data model. Moreover, different vocabularies are introduced to unambiguously represent these entities. Finally, the Observable object is extended by an attribute for the unstructured capture of event data. Table 2 gives an overview of all these adjustments to the data format.A detailed overview of the specific extensions integrated as well as the actual object specifications can be found in the repository published together with this

Table 2 STIX extensions

Classification	Base entity	Additional Entity	Additional attribute
Intelligence	TTP	Structural Event	StructuralEventTypeVocab
	TTP	Environmental Event	EnvironmentalEventTypeVocab
	TTP	Legal Event	LegalEventTypeVocab
Attribution	Threat Actor	Structural Source	StructuralSourceTypeVocab
	Incident		
Indicator	Observable		Observation

work[4]. The repository includes XML-schema definitions for the STIX schema extension types and vocabularies that are developed with this work.

4.5 Structured acquisition of human-as-a-security-sensor information

To implement a system harvesting incident information from a human sensor, it is necessary to develop a systematic approach to guide the user through the acquisition (Q3). This supports the structured input into a data format 4.4 and encourages human sensors to provide as much information as possible. The process for guiding the user is basically given by the IHaaSS Incident Model and Taxonomy as shown in Fig. 3. Thereby, multiple threat sources, threat events, and entities can be specified consecutively. The expected impact is estimated for the whole incident and thus recorded only once. The respective subtypes for sources, events or entities are also gathered in hierarchical sequence to avoid overstraining of the user. Each event is assigned a cause (either a threat source or another threat event), which leads to a chain of events. However, the process is subject to some constraints. More precisely, threat events cannot be initiated by some threat sources or preceding threat events. The constraints for our acquisition process are defined as follows and explained in more detail subsequently. The notation is based on the formal model of Klingner and Becker (2012):

$$prohibits(Attack) = Environmental \vee Legal \tag{1}$$

$$prohibits(Technical) = Legal \tag{2}$$

$$prohibits(Environmental) = Legal \tag{3}$$

$$\begin{aligned} prohibits(UnusualNaturalEvent) \\ = Actor \vee Structure \vee Attack \\ \vee Technical \vee Legal \end{aligned} \tag{4}$$

Equation 1 defines that an attack cannot be initiated by an environmental or a legal event. The reason for this is that an attack requires action by a human being or at least some technical device and thus cannot be initiated by nonhuman events or sources. Furthermore, the cause of a technical security event cannot be a legal event (Eq. 2), *technical events* can only follow *physical events* or *sources*. The same applies to environmental events (Eq. 3). *Unusual natural events* (e.g. sunspots) cannot be caused by any other events or sources except *Environmental* ones as stated in Eq. 4, because they have a natural cause.

These constraints are the most explicit ones. It would be possible to define additional constraints considering more detailed layers of the underlying taxonomy. However, the constraints would depend on the organization where they are implemented and would not be unambiguous.

5 Evaluation

In the previous sections, we presented an approach for integrating human sensor information into SIEM systems. Therefore, we first discussed possible connecting points for the interaction between human sensors and SIEM systems. We also developed an incident model that extends the scope of SIEM threat detection by incidents that are additionally detectable by human sensors. Based on these findings, we extended the STIX data model to create data structures capable of capturing this information and proposed a concept for the structured acquisition of human sensor information. In design science research, demonstration is like a light-weight evaluation, to show that the artifact works to solve instances of a given problem (Venable et al. 2012; Peffers et al. 2007). To evaluate that our approach achieves its purpose in our context, we demonstrate it threefold: First, we explain our prototypical implementation, which shows that it is realizable in practice. Thereafter, we use the example from chapter 2 to show that it can be mapped to the IHaaSS Incident Model and Taxonomy presented in Section 4.2. Finally, we demonstrate how this example would be represented in the STIX based format presented in 4.4. Hereby it is worth mentioning, that a taxonomy is never perfect and has to be shaped and extended as the field of its purpose advances (Nickerson et al. 2013). Furthermore, it is hardly possible to evaluate the taxonomy going beyond a demonstration, since it can only

[4] http://tinyurl.com/y3h5k25t

be shown exemplary, that it fits its intended purpose. This is especially true for the context of this paper, as there are almost no limits to the variety of cyberattacks and incidents. To the best of our knowledge, there is no similar taxonomy describing the artifacts that can be recognized by human security sensors. Thus it is not possible, to compare the performance of our taxonomy to others.

5.1 Prototypical implementation

Our application prototype realizes the rendering of information delivered by human security sensors into the structured threat intelligence information. A working example of the IHaaSS prototype is available online[5]. The prototype pursues two different goals. On the one hand, it demonstrates the use of IHaaSS in a possible scenario for the structured acquisition of incident information to show the overall validity of our approach. On the other hand, it illustrates the value of information delivered by human security sensors and the combination possibilities with data from existing analytics processes. The application consists of two major components: First, a wizard component that allows the reception of incident information delivered by humans. Second, a server component that translates the acquired incident information into the structured format to be further processed afterwards. The frontend is implemented by using Angular[6] and Typescript[7]. Java EE in combination with a Glassfish[8] application server was used to implement the STIX conversion logic and the database access.

Figure 4 shows a screenshot of the first step in the wizard component. The wizard is divided into two components. In the first component, the information can be entered by the user. The second part (Captured elements) gives an overview of already declared incident elements so that the user can see what has been previously entered. The wizard is structured in four steps as specified by the taxonomy. At first, the threat sources can be reported. Thereby, an arbitrary number of sources can be added. For selecting a source, the user is presented a drop-down list containing the elements of the first layer of the taxonomy (Actor and Structure). When an element is selected, a second drop-down list with the elements of the next layer is displayed. This continues until there are no sub-elements left. The same selection mechanism is implemented for event types and entities in subsequent steps. Only for events a "triggered by" input field is added to specify the previously reported threat source or threat event that initiated the event. There the selectable events get filtered according to the constraints defined in chapter 4.5. In the fourth and final step, the estimated impact of the whole incident can be entered. Furthermore,the following additional information is requested:

- **Email:** The email is used to enable follow-up contact to the user who reported an incident for example when additional information is required.
- **Date:** The date on which the incident occurred. The current date is used as default value.
- **General description of the incident:** A free text explanation of the incident enables the statement of additional context information.
- **Technical data:** This input field is used for providing technical information like log data or the content of a phishing mail. This information could also be gathered partially automatically as described by Heartfield and Lukas (2018) depending on the incident and the organizations' infrastructure.

After the incident information was acquired by the wizard component, the data is transferred to the backend component. The backend provides the conversion logic, which translates the information collected by the wizard into corresponding STIX objects. It also provides the underlying data storage for persisting the translated STIX objects for later use.

5.2 Case study

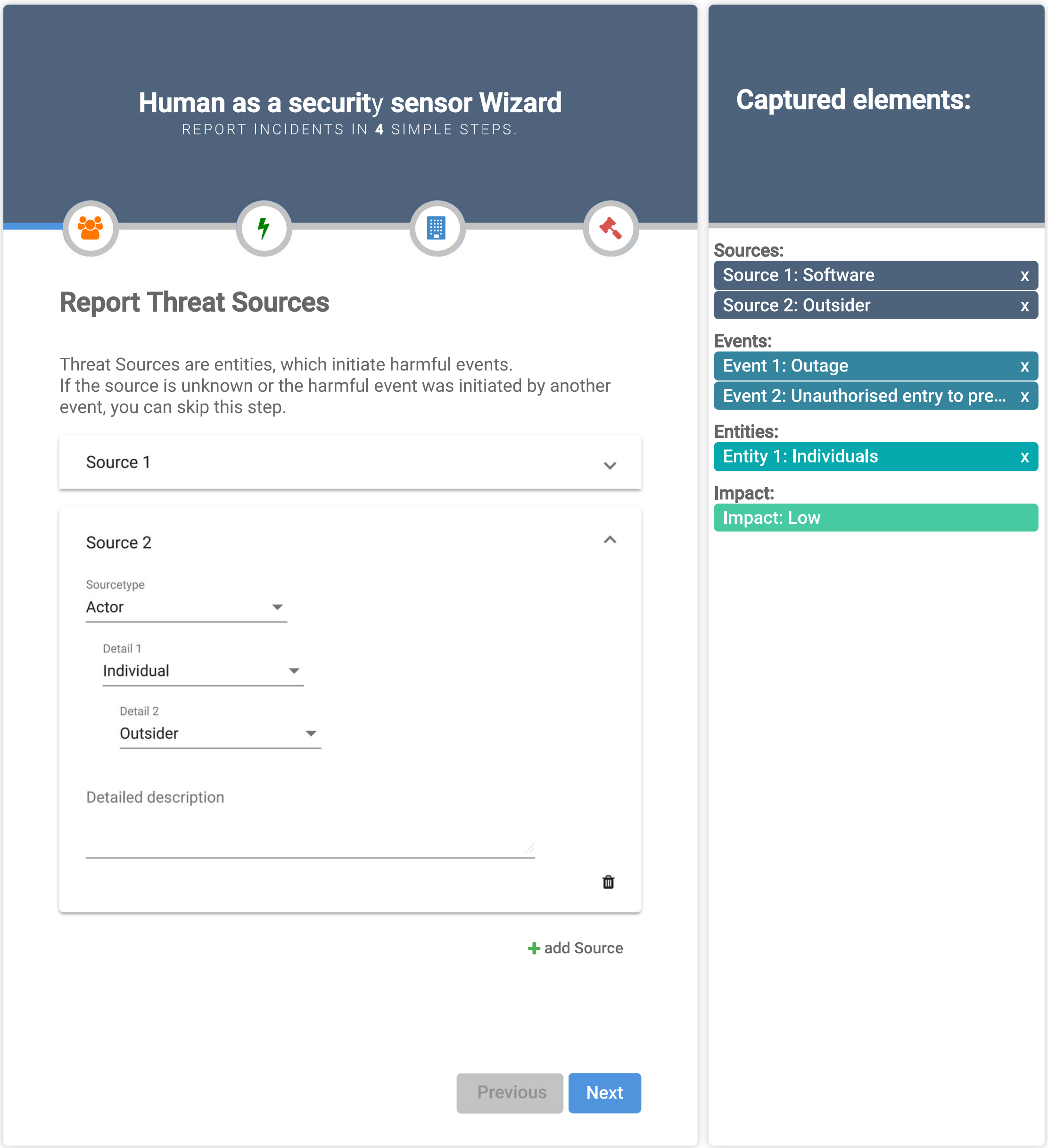
In order to evaluate the wizard in combination with the underlying taxonomy and constraints we show how an employee could report an incident using the wizard. We used the DarkVishnya incident as shown in Section 2 as an exemplary use-case, which we iterate through below. Note that we take the role of a fictional employee that could have observed the incident. Thus, we only consider occurrences that may have been observed by a non-technical staff member for this example. The potential selection steps within the wizard are subsequently shown in brackets. For this incident, we identified the following two threat sources:

1. An unknown person is observed inside the premises (Actor → Individual → Outsider)
2. A suspicious hardware device is seen in an office room (Structure → IT Equipment → Processing)

Moreover, two threat events can be identified:

1. The person falsely claims to have legitimate access and enter s the premises (Attack → Physical attacks → Unauthorized entry to premises)

[5] http://tinyurl.com/yyqqlgg7
[6] https://angular.io/
[7] https://www.typescriptlang.org/
[8] https://javaee.github.io/glassfish/



Human as a security sensor Wizard

REPORT INCIDENTS IN **4** SIMPLE STEPS.

Report Threat Sources

Threat Sources are entities, which initiate harmful events.
If the source is unknown or the harmful event was initiated by another event, you can skip this step.

Source 1

Source 2

Sourcetype: Actor

Detail 1: Individual

Detail 2: Outsider

Detailed description

Captured elements:

Sources:
- Source 1: Software x
- Source 2: Outsider x

Events:
- Event 1: Outage x
- Event 2: Unauthorised entry to pre... x

Entities:
- Entity 1: Individuals x

Impact:
- Impact: Low

Fig. 4 Screenshot of the wizard for reporting incidents by humans

2 The hardware device is placed in an office room and connected to internal network infrastructure (Attack → Nefarious activity/Abuse → Manipulation of hardware and software)

In addition, a network device was identified as a negatively affected entity. Thus, assets are selected from the wizard. Finally, the impact is estimated as low, since the employee may not be able to judge the whole extent of the incident. After the data is collected from the human sensor, it is translated into the corresponding STIX data objects by the server component as described in the following. The outsider (1) who falsely claimed to have legitimate access to the premises is translated into a Threat Actor object. Its specific properties are mapped to the internal vocab "ThreatActorTypeVocab" that was extended within this work. The technique of gaining unauthorized access to the premises is translated into a TTP object and matching attack patterns from the CAPEC enumeration are mapped. The suspicious hardware device (2) attached to the internal network is then mapped to a structural source object and its specifics are mapped using the "StructuralSourceTypeVocab" created with this work. The action of planting a malicious device is described using a further TTP object and the corresponding CAPEC attack patterns analogous to the first TTP object. After creating these specific entities, the general descriptions of the incident as well as the time of the occurrence, affected assets, and the expected impact are recorded using an Incident object. All these objects are then finally wrapped using a Report object. The complete STIX report for this exemplary use-case is appended to this work as supplementary material. Moreover, it can be viewed under the past incidents overview section within the wizard prototype[9].

Considering the results of this incident, there are different possible connecting points to automated analyses within a SIEM system. Firstly, the generated report delivers information about the approximate time of the occurrence, the exact location as well as the affected network device and possibly even the used network port. This data can then be enriched with the corresponding log information from the SIEM system in order to clarify the findings. Furthermore, if an electronic access control has been circumvented in any way, the log data available can also be used as further evidence and to enrich the incident information gathered from the employee.

5.3 Discussion

The prototypical implementation has shown three key aspects: First, it was demonstrated, that it is possible to represent the beforehand theoretically defined IHaaSS incident model and taxonomy (Section 4.2) as a wizard-like application. This application guides the user through the taxonomy and enables him to select and report all possible elements. Second, the acquisition can be conducted in a structured way since the constraints defined in Section 4.5 were all implemented within the prototype. Nevertheless, practical usage over a longer period of time will reveal whether these constraints are exhaustive. Third, the acquired data can be translated into a STIX representation, which could be further used for security analytics systems, despite the volume of possible user input.

The case study has shown that it is generally possible to apply the prototype for a real-word incident. Therefore, it was validated with an expert who analyzed the attack as a member of the incident response team. However, only a broad long-term study can show the usability, which we will address in the future.

6 Conclusion and future work

In this paper, we present an approach for acquiring and structuring incident information from human sensors to prepare it for the use within security analytics systems such as SIEM systems. Therefore, we identify the connection points of human sensors within a SIEM system (Q1) and answer the question how the reportable information can be structured (Q2). Thereby the IHaaSS Incident Model and Taxonomy is deduced, which consists of the four components threat sources, threat events, entities and expected impact. The incident model builds the basis for a data format suitable for representing threat intelligence information reported by humans. An important factor while developing the data format is to maintain the compatibility with existing and well-established formats, in our case STIX. For acquiring the data from human sensors in a structured way (Q3) we propose a process where we define some constraints, which ensure that the collected data is not contradictory. Finally, the approach is evaluated from three directions. First, we prototypically implement the approach and second, an example use-case is mapped to the IHaaSS Incident Model and Taxonomy to show its practicability. Finally, the use case was represented in the proposed STIX-based format.

Since the examined subject of human-as-a-sensor, especially with its focus on security, is a rather new topic, there is a lot of potential for future research. A topic marginally tackled in this paper is the connection of human-generated data with machine-generated data, which for example originates from log files. The data collected from humans may be extended by automatically or manually deriving relationships to machine data. To achieve this, different approaches such as rule-based correlation and aggregation may be used. In order to facilitate the definition of rules, it can be helpful to visualize the generated data. Therefore, existing approaches as presented by Böhm et al. (2018) could be extended to the proposed data format. Machine learning techniques also show a lot of potential regarding the correlation of data acquired by humans and machine-generated data.

Our present work considers the acquisition and structuring of information delivered by humans. However, we have not examined forensic and legal requirements. Nevertheless, considering these requirements is of great relevance especially when the collected data is supposed to be used as evidence in court afterwards. Furthermore,

[9] http://tinyurl.com/y5tsoxo3

human generated data may also play an important role in the incident response process and thus should be qualified as data foundation for forensic analyses. Since reports may contain personal data, the topic needs additional consideration from a legal point of view.

An additional research gap can be identified with regard to motivating employees for reporting detected incidents. On the one hand, incentives have to be created and on the other hand, barriers keeping employees from reporting have to be removed. For example, if a person reports an incident, which denigrates a colleague, it might be an unwanted result. In this context, obfuscation techniques, such as anonymization or pseudonymization, may help to solve some of these problems. Additionally, changes to the corporate culture are required, so that it is considered normal for employees to report detected incidents, as it is for example in an anti-fraud culture. In this regard the analysis and assurance of data quality is especially important due to the possibility of erroneous inputs by humans. Finally, the proposed approach is rather generic. Thus, it has to be adopted to the respective context for practical use. Especially the proposed taxonomy could be refined in order to depict more corporate information and it has to be tailored to match the corporate culture.

Acknowledgements
This research was supported by the Federal Ministry of Education and Research, Germany, as part of the BMBF DINGfest project (https://dingfest.ur.de). We also thank Sergey Golovanov of Kaspersky Lab for providing us with detailed information about the Dark Vishnia incident, which significantly enhanced this paper.

Funding
Not applicable.

Availability of data and materials

Source code - iHaaSS wizard
Project name: Client
Project home page: http://tinyurl.com/y6kjrb4q
Archived version: 1.0
Operating system(s): Platform independent
Programming language: HTML, Typescript/JavaScript
Other requirements: Apache Webserver or similar, NPM 6.2.0 or higher
License: GNU GPL v3

Source code - sTIX server
Project name: STIX Server
Project home page: http://tinyurl.com/y46hsvj8
Archived version: 1.0
Operating system(s): Platform independent
Programming language: Java EE 7
Other requirements: Glassfish version 4.1.1 or higher
License: GNU GPL v3

Additional sTIX schema files
Project name: STIX-Schema
Project home page: http://tinyurl.com/y2s3ba7k
Archived version: 1.0
Operating system(s): Platform independent
Programming language: xml-schema
License: BSD-3-Clause
Appended as supplementary material

Received: 24 April 2019 Accepted: 29 August 2019
Published online: 22 October 2019

References

Anti-Phishing Working Group I Report Phishing. https://www.antiphishing.org/report-phishing/overview/. Accessed 19.01.2019

Appala S, Cam-Winget N, McGrew D, Verma J (2015) An Actionable Threat Intelligence system using a Publish-Subscribe communications model. Proc 2nd ACM Workshop Inf Sharing Collab Secur - WISCS '15:61–70

Barnum S (2014) Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX). http://stixproject.github.io/getting-started/whitepaper/. Accessed 2019-02-21

Bhatt S, Manadhata PK, Zomlot L (2014) The operational role of security information and event management systems. IEEE Secur Privacy 12(5):35–41

Böhm F, Menges F, Pernul G (2018) Graph-based visual analytics for cyber threat intelligence. Cybersecurity 1(1)

Burger EW, Goodman MD, Kampanakis P, Zhu KA (2014) Taxonomy model for cyber threat intelligence information exchange technologies. In: WISCS '14 Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security Vol. WISCS '14. pp 51–60

Crowley C, Pescatore J (2018) Sans 2018 security operations center survey

Dandurand L, Kaplan A, Kácha P, Kadobayashi Y, Kompanek A, Lima T, Millar T, Nazario J, Perlotto R, Young W (2015) Standards and Tools for Exchange and Processing of Actionable Information

Fenz S, Heurix J, Neubauer T, Pechstein F (2014) Current challenges in information security risk management. Inf Manag & Comput Secur 22(5):410–430

Fransen F, Smulders A, Kerkdijk R (2015) Cyber security information exchange to gain insight into the effects of cyber threats and incidents. Elektrotechnik & Informationstechnik 18:106–112

Google LLC. Gmail. https://mail.google.com/. Accessed 19.01.2019

Heartfield R, Loukas G, Gan D (2016) You are probably not the weakest link: Towards practical prediction of susceptibility to semantic social engineering attacks. IEEE Access 4:6910–6928

Heartfield R, Loukas G (2018) Detecting semantic social engineering attacks with the weakest link: Implementation and empirical evaluation of a human-as-a-security-sensor framework. Comput Secur 76:101–127

Hintzbergen J, Hintzbergen K, Smulders A, Baars H (2015) Foundations of Information Security: Based on ISO 27001 and ISO 27002. 3rd. Van Haren Publishing, Zaltbommel

Holik F, Horalek J, Neradova S, Zitta S, Marik O (2015) The deployment of security information and event management in cloud infrastructure. In: 2015 25th International Conference Radioelektronika (RADIOELEKTRONIKA). pp 399–404

ISO/IEC 27001: Information technology – Security techniques – Information security management systems – Requirements (2013) Technical report. Int Org Standard

Joint Task Force Transformation Initiative (2012) Guide for Conducting Risk Assessments. National Institute of Standards and Technology, Gaithersburg, MD

Juliadotter NV, Choo K-KR (2015) Cloud attack and risk assessment taxonomy. IEEE Cloud Comput 2(1):14–20

Klingner S, Becker M (2012) Formal modelling of components and dependencies for configuring product-service-systems. Enterp Model Inf Syst Architectures 7(1)

Kostakos V, Rogstadius J, Ferreira D, Hosio S, Goncalves J (2017) Human sensors. In: Participatory Sensing, Opinions and Collective Awareness. Springer, Cham. pp 69–92

Lineberry S (2007) The human element: The weakest link in information security. J Account 204(5):44

Marinos L (2016) ENISA Threat Taxonomy: A Tool for Structuring Threat Information

Mello J (2017) Security Awareness Training Explosion. https://cybersecurityventures.com/security-awareness-training-report/. Accessed 28.02.2019

Menges F, Pernul G (2018) A comparative analysis of incident reporting formats. Comput Secur 73:87–101

Microsoft Corporation Deal with abuse, phishing, or spoofing in Outlook.com. https://support.office.com/en-us/article/deal-with-abuse-phishing-or-spoofing-in-outlook-com-0d882ea5-eedc-4bed-aebc-079ffa1105a3

Nickerson RC, Varshney U, Muntermann J (2013) A method for taxonomy development and its application in information systems. Eur J Inf Syst 22(3):336–359

Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S (2007) A design science research methodology for information systems research. J Manag Inf Syst 24(3):45–77

Rahman SS, Heartfield R, Oliff W, Loukas G, Filippoupolitis A (2017) Assessing the cyber-trustworthiness of human-as-a-sensor reports from mobile devices. In: 2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA). pp 387–394

Shackleford D, SANS Institute (2015) Who's Using Cyberthreat Intelligence and How? https://www.alienvault.com/docs/SANS-Cyber-Threat-Intelligence-Survey-2015.pdf. Accessed 2019-02-21

Sauerwein C, Sillaber CN, Mussmann A, Breu R (2017) Threat intelligence sharing platforms: An exploratory study of software vendors and research perspectives. In: 13. Internationale Tagung Wirtschaftsinformatik, WI 2017, St. Gallen

Golovanov S (2018) DarkVishnya: Banks attacked through direct connection to local network. https://securelist.com/darkvishnya/89169/

Turnbull J (2019) The Art of Monitoring. Version 1.0.4

Venable J, Pries-Heje J, Baskerville R (2012) A comprehensive framework for evaluation in design science research. In: International Conference on Design Science Research in Information Systems. pp 423–438

Vielberth M, Pernul G (2018) A security information and event management pattern. In: 12th Latin American Conference on Pattern Languages of Programs (SugarLoafPLoP 2018)

Wang D, Amin MT, Li S, Abdelzaher T, Kaplan L, Gu S, Pan C, Liu H, Aggarwal CC, Ganti R, Wang X, Mohapatra P, Szymanski B, Le H (2014) Using humans as sensors: An estimation-theoretic perspective. In: IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks. IEEE, Piscataway. pp 35–46

Publisher's Note

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

5 Towards GDPR-compliant data processing in modern SIEM systems

Current status:	Under Review
Journal:	*Submitted to:* Computers & Security
Date of acceptance:	n/a
Full citation:	Florian Menges , Tobias Latzo , Manfred Vielberth , Sabine Sobola , Henrich C. Pöhls , Benjamin Taubmann , Johannes Köstler , Alexander Puchta , Felix Freiling , Hans P. Reiser and Günther Pernul. Towards GDPR-compliant data processing in modern SIEM systems. *Working Paper, University of Regensburg, 2020.*

Authors contributions:	Florian Menges	20%
	Tobias Latzo	15%
	Manfred Vielberth	15%
	Sabine Sobola	10%
	Henrich C. Pöhls	10%
	Benjamin Taubmann	5%
	Johannes Köstler	5%
	Alexander Puchta	5%
	Hans P. Reiser	5%
	Felix Freiling	5%
	Günther Pernul	5%

Journal Description: Computers & Security is the most respected technical journal in the IT security field. With its high-profile editorial board and informative regular features and columns, the journal is essential reading for IT security professionals around the world.

Towards GDPR-compliant data processing in modern SIEM systems

Florian Menges[1], Tobias Latzo[2], Manfred Vielberth[1], Sabine Sobola[4], Henrich C. Pöhls[3], Benjamin Taubmann[3], Johannes Köstler[3], Alexander Puchta[1], Felix Freiling[2], Hans P. Reiser[3], and Günther Pernul[1]

[1] Department of Information Systems, Universität Regensburg, Germany
[2] Department of Computer Science, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
[3] Institute of IT-Security and Security Law, Universität Passau, Germany
[4] Paluka Sobola Loibl & Partner attorneys at law, Regensburg, Germany

Abstract. The introduction of the General Data Protection Regulation (GDPR) in Europe raises a whole series of issues and implications on the handling of corporate data. We consider the case of security-relevant data analyses in companies, such as those carried out by Security Information and Event Management (SIEM) systems. It is often argued that the processing of personal data is necessary to achieve service quality. However, at present existing systems arguably are in conflict with the GDPR since they often process personal data without taking data protection principles into account. In this work, we first examine the GDPR regarding the resulting requirements for SIEM systems. On this basis, we propose a SIEM architecture that meets the privacy requirements of the GDPR and show the effects of pseudonymization on the detectability of incidents.

Keywords: Security Information and Event Management · SIEM · GDPR · Threat Intelligence · DINGfest.

1 Introduction

1.1 Motivation

The security of the modern information infrastructure is of high importance. In order to detect misuse and attacks at an early stage a lot of information about the events inside IT-infrastructures, e.g. inside computer networks and software applications also across many systems, is required to detect or post-mortem report and document attacks. *Security Information and Event Management* (SIEM) systems help organizations to keep up with the ever increasing complexity by providing a holistic view on IT-infrastructures. Naturally, SIEM systems process enormous amounts of data about security related events, e.g., when specific users login or certain users perform critical actions. It is often argued, that generally the quality of service depends critically on the quality and detail of the data collected and processed within the system [48], which has been shown for different domains such as threat intelligence [37].

Events like those just described that are processed within the SIEM system are clearly related to concrete users and therefore must be treated as personal information, which require protection under Europe's *General Data Protection Regulation* (GDPR) [13]. Adopted in May 2018, it regulates and harmonizes the protection of personal data in the processing and transfer of data within and between private companies and/or public bodies in the European member states. Although, the GDPR is only compulsory for EU member states, it has evolved into a blueprint for data protection all over the world, as discussions between the US Congress and Mark Zuckerberg in the aftermath of the Cambridge Analytica case indicate[5].

Hence, SIEM systems must also comply to the regulations themselves, which leads to conflicting interests. On the one hand, SIEM systems rely on personal data such as information from the identity and access management (IAM) for providing high detection rates of incidents and thus a high level of protection. On the other hand, the requirements of the GDPR suggest that investigations of data streams as carried out in current SIEM systems may no longer be legally compliant. To complicate things even further, regulations regarding the handling of digital evidence mandate that authenticity and integrity of the data related to an incident should be guaranteed at all times in order to maintain its high legal probative value. It is therefore necessary to find the best trade-off between those two demands. With this work we attempt to fill the resulting research gap and to harmonize legal GDPR requirements with the technical architecture for SIEM systems. To bridge the gap between the disciplines of computer science and law and to produce the most reliable results possible, this paper was written by IT security researchers in collaboration with a lawyer A central idea is the integration of *anonymization* and *pseudonymization* into threat analytics mechanisms. While this makes it necessary to change the original data, it is possible to maintain legal integrity and authenticity by using *redactable and sanitizable signatures*, a cryptographic concept to retain a level of authenticity useful to retain a suitable level of legal evidence even when data gets obfuscated or if certain parts of it are missing. We deploy cryptography to enable balancing authenticity proofs for the collected security-related events with the confidentiality requirements of the information about commercially-relevant internals (trade secrets) and employees' as well as customers' privacy (personal data). Thus, our goal is to minimize the amount of data which is being made accessible to third-parties in every step of the SIEM process. By enforcing this with cryptography the proposed system adheres to the security-by-design principle of least privilege as well as the privacy-by-design principle of data minimization. At the same time we aim to keep the impact on detection as low as possible and thus we provide an audit-able process to gain access to more details if security analysis is needing

[5] https://www.theverge.com/2018/4/11/17224492/zuckerberg-facebook-congress-gdpr-data-protection

it. For the reason of being able to reconstruct original data, leaving a trace in an audit log, we focus on cryptographic methods and support pseudonymization rather than anonymization. Technically, we encrypt and sign events early and store the decryption keys with a party trusted for logging access to stored keys; moreover we employ signatures that allow to slice or redact data.

1.2 Related Work

When looking at the application of privacy mechanisms to threat analytics (e.g. SIEM systems), literature can be divided into a pre-GDPR and a post-GDPR phase, as this regulation still has a big impact on the integration of privacy. In the former phase there are not many results to be found regarding applying privacy to SIEM systems, however the challenges in integrating privacy in forensic and threat analyses has been identified [40,17]. Although the challenges were not solved for SIEM systems, selected works in the IT security domain address it. For example Burkhart et al. [6] describe a privacy preserving solution for secure multi-party computation. Furthermore, a main focus during this era was the application of privacy to intrusion detection systems (IDS), which could be declared as the predecessors of modern SIEM systems and thus in our context are worth a closer look: Sobirey et al. [39] propose an approach for pseudonymizing user related data in IDS and closely examined, which records need to be pseudonymized in audit records. Based on this work, Biskup and Flegel [3] and Park et al. [28] propose an approach which is quite similar to the one presented in this paper as it uses cryptograpic methods to pseudonymize personal data, though these are closely tailored to IDS and not completely adaptable to SIEM. In addition, they were issued before the publication of the GDPR and thus did not have all the requirements in mind and respectively were not evaluated against the new requirements. Our approach also differs,as we have a more abstract view of the whole system and do not focus largely on cryptographic details. Furthermore, Buschkes and Kesdoğan [7] discuss requirements such as data avoidance and reduction of personal data.

Although privacy preserving methods were widely discussed in the past, recently the application of GDPR received an increased amount of attention and new works were published. In relation to SIEM, some work was published covering GDPR compliant data processing. Sgaglione and Mazzeo [38] and Coppolino et al. [8] introduce the COMPACT project, which is a GDPR compliant SIEM. However, they do not go into detail, how this is realised technically. Current research for SIEM systems mainly focuses on the architecture and improvement of such systems and not on the integration of privacy [26,25,27].

In cryptography, digital signatures are used to ensure authenticity and integrity of data, i.e., they guarantee that upon inspection data is unchanged and comes from an attributable source. Special techniques of *redactable signature schemes* (RSS) by Steinfeld et al. [41,18] allow subsequent deletions in the data, while *sanitizable signature schemes* (SSS) as proposed by Atieniese at al. [1] even allow subsequent edits by dedicated authorised parties while maintaining authenticity of the remaining data. Both RSS and SSS allow to balance authenticity with privacy protection, because they allow retaining the integrity and authenticity protection for the unedited or not-removed parts of the document and at the same time keep the confidentiality protection for the overwritten parts of the document. In cryptography the latter property is intuitively termed privacy. While many schemes have appeared in the literature [2], only some of which uphold privacy and only those schemes that additionally fulfil detectability, known as non interactive public accountability [5] can be used for eIDAS[6] compliant signatures [30,31,45]. While the legal compliance of such signatures has been subject to research, the integration of such schemes into privacy protection of SIEM have not yet been investigated. In particular, in the application scenario of SIEM we want to be able to later reveal previously not-shared content. For this, a special form of digital signatures is needed which has the property of *mergeability*, i.e., the ability to re-add signed content to previously redacted but still signed content and re-generate a valid signature over the merged content [34].

1.3 Contribution and Outline

To the best of our knowledge we are not aware of an approach, that integrated GDPR into SIEM in a comprehensive way. Given the fact that these regulations need to be applied by all companies that operate within the European Union, there appears to be high demand for systems that are GDPR compliant. In this paper we present the first privacy-friendly – and thus GDPR-compliant – SIEM architecture that protects the confidentiality as well as the authenticity of security-relevant events starting at their collection, keeping the protection during the analysis and finally sending an incident report.

The presented architecture allows the deployment of a SIEM that meets the regulatory requirements under the EU data protection. It protects personal information in the data sets from unnecessary visibility using pseudonymization and encryption techniques without a significant reduction in detectability. Hence, we balance data-quality (detection of incidents) with legal obligations from privacy legislation and thus also protect trade secret by sharing only the minimum necessary information in any step of the SIEM process. Thus we strongly adhere to the GDPR's data minimization principle. Still, we achieve the highest level of confidence that the security-relevant events initially recorded and reported into the SIEM process are protected from tampering by using redactable and sanitizable signature schemes to proof authenticity. This allows us to balance the need for generating data with a high legal evidence with the need to protect privacy (and trade-secrets).

The legal analysis carried out for this architecture and presented in this paper shows that even potentially invasive data can be collected in a GDPR-compliant manner as our proposed system balances the necessity of the collection (detection and reporting of actual security incidents) with the protection of users

[6] eIDAS is short for the current legislation which defines the technical functionalities to allow electronic signatures to be legally equivalent to handwritten signatures within the EU [12].

privacy and customer's trade-secret needs. The paper shows the actual influence of pseudonymization on incident detection mechanisms and the results of the performed legal evaluation.

The remainder of this paper is structured as follows. First, we give some background on the GDPR (legal) and SIEM (technical) in Section 2. In Section 3 we develop the research questions that arise from integrating GDPR into SIEM. On this basis, we describe our GDPR-compliant architecture for SIEM systems in Section 4. The architecture is the evaluated on both technical and legal level in Section 5. The paper concludes in Section 6.

2 Background

This section provides the background information that is needed to understand the approach presented in this paper. Thus, we first give an overview of the functionality and properties of SIEM systems, as this serves as a basis for our architecture. Subsequently, we give an overview of the requirements the GDPR defines with special attention to the processing of personal data.

2.1 Security Information and Event Management (SIEM)

In general, SIEM was first mentioned by Gartner [49]. It originated from the initially separate systems Security Information Management (SIM) and Security Event Management (SEM) [15]. SIEM must fulfill several requirements, which are all connected: Log collection, enrichment with context data, log normalization, event correlation, and analysis as well as long- and short-term storage of log data, reporting, monitoring, alerting, and incident response [24,47,14].

A SIEM system as described in [47] is essentially designed for collecting relevant log data in a central place from arbitrary systems such as network devices or operating systems. This among other things enables the detection of incidents and in this way gaining situational security awareness. On a high level of abstraction, a SIEM system consists of the three main steps *data acquisition*, *processing* and *reporting*, which are elaborated in the following in more detail.

Data acquisition: Hereby, it first collects relevant event information, in most cases in the form of log data, which gets enriched with additional context data. There are basically two approaches for data acquisition: First, the data can be pushed into the SIEM by the data generating system. Thereby, the SIEM does not influence the generated data. Second, the data can be pulled by the SIEM from the observed system, which grants more control over the generated data enabling for example the assurance of integrity. This data then is translated into a uniform representation during the normalization step.

Processing: The core of the system is the correlation and analysis component, wherein information from various sources is correlated and incidents get detected by methods such as pattern matching. Real-time threat detection enables fast reactions in case of an incident, whereas forensic analysis pursues the goal of analyzing the whole extent of the event in the aftermath in order to secure evidence. A distinction can, therefore, be made between short-term and long-term storage of relevant data. For long-term storage, it is particularly important to preserve the data in a tamper-proof way in order to be able to use it as evidence in court. Monitoring and visual security analytics enable security analysts to be actively involved in the analysis process. In the case of a detected incident, alerting and incident response triggers necessary reactions to mitigate further harm.

Reporting: An essential part of modern SIEM systems is reporting occurred incidents for compliance reasons (e.g. critical infrastructure providers) or enables the participation in established threat intelligence sharing platforms between participating organizations.

2.2 GDPR

In the following we present some background on general and SIEM-specific GDPR demands and later (see Sect.3) detail, which problems arise for SIEM systems to be built to comply.

Since 25 May 2018, the GDPR has been in force throughout the European Union (EU) to ensure the protection of the "natural person" in data processing. This regulation is directly applicable to all member states of the EU. The GDPR does not contain any immediate legal requirements for software developers but if they intend to sell their product to customers, they must be aware of the legal requirements.

Data protection is the protection of the natural person from privacy impairments through the processing of data concerning the person. Everyone should be free to decide who, when and how their data should be accessible. The term of personal data is therefore defined as "all information relating to an identified or identifiable natural person", Art. 4 (1) GDPR.

General principles of data processing In order to achieve the goal of high personal protection, the GDPR pursues the regulation of all basic principles which are regulated in Art. 5 (1) GDPR. According to the GDPR, only lawful, fair and transparent data processing is permitted in a transparent manner, Art. 5 (1) lit. a GDPR, in order to serve the principle of *good faith*. Furthermore, data processing shall be lawful only if and to the extent that it is applied by Art. 6 GDPR. But even then, it must be done in a transparent way which is traceable for the data subject. Another important principle is the *purpose limitation*, Art. 5 (1) lit. b GDPR. Thus, the clear purpose of processing must be previously established and legitimate. Furthermore, the *principle of data minimization* is applicable, Art. 5 (1) lit. c GDPR. This means that data collection is only allowed within limits for specified, explicit and legitimate purposes and not further. A further principle is *accuracy*, Art. 5 (1) lit. d GDPR. Only correct data may be collected. Even after processing it must be ensured that personal data is accurate. If this is not the case, data must be erased or rectified with delay. One further fundamental principle is the storage limitation, Art. 5 (1) lit. e GDPR. In the course of data processing it must be ensured that an identification is only possible in case of it being necessary for the purpose. *Integrity and confidentiality* have recently become further important principles, Art. 5 (1) lit. f GDPR. This means that processing must occur in compliance with general safety standards. Compliance with these principles must be proven at any time by the controller, Art. 5 (2) GDPR.

Processing on a legal basis and transparency obligations Generally, there must be a legal basis for the processing, otherwise all collection of personal data is considered unlawful. The exceptions are regulated by Art. 6 GDPR. First of all, processing is lawful if the data subject has given *consent* to data processing, Art. 6 (1) point (a) GDPR. Processing is also allowed for the performance of contracts, Art. 6 (1) point (b) GDPR. If the processor is subjected to a legal obligation, data processing is also lawful without further requirements, Art. 6 (1) point (c) GDPR. This also applies if the protection of vital interests is pursued, Art. 6 (1) point (d) GDPR. The processing is also possible for the performance of a task carried out in the *public interest* or in the *exercise of official authority*, Art. 6 (1) point (e) GDPR. Lastly, processing is lawful for the purposes of the *legitimate interests pursued by the controller or by a third party*, except where such interests are overridden by the interests or fundamental rights and freedoms of the data subject which require protection of personal data, Art. 6 (1) point (f) GDPR. While a lot of the exceptions might be triggered by the need and want to have a SIEM to protect from security breaches, the collection must meet a balance test, e.g. collection must not overshoot the goal, and must be transparent, e.g. clearly communicated to the data subjects.

Data security In addition, the GDPR regulates a close interconnection of data protection with data security (especially Art. 32 GDPR) to the effect that technical and organizational measures in the data-processing company enable the highest degree of data security (availability, confidentiality, integrity, and resilience). In our scenario, the two security goals confidentiality and integrity are particularly relevant. We must ensure that information about events that could relate to incidents is transferred from the source to the sink, and is not altered or made accessible to unauthorised persons. To protect integrity an unauthorized modification must be detected if it happened, which is especially important to use non-tampered recorded data as evidence. Thus, protecting integrity and authenticating the data's origin provides legal value. Further, confidentiality protection guarantees that no unauthorized party is able to obtain information not intended for them, e.g. we must securely communicate the personal data to have them reach only the right recipients.

Redaction The term "redaction" itself is not found in the GDPR directly; it refers to the irreversible removal of the information [43]. This process is explicitly mentioned in guidance documents that explain how to remove information that is not subject to the information to be released under laws for the freedom of access to information, e.g. UK FOIA [29] and thus is also applicable as a technique in the context of GDPR's data minimization [44,43].

Pseudonymization The term "pseudonymisation" of the GDPR means the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organizational measures to ensure that the personal data are not attributed to an identified or identifiable natural person, Art. 4 (5) GDPR. In order to make a pseudonymization, the data subject must first be assigned to pseudonyms. These can be user IDs. Thereafter the necessary data for identification must be kept separately. It must be ensured that they are strictly separated from the pseudonyms. It is important that features that can only indirectly lead to identification must be removed in the event of pseudonymization. The pseudonymization can be made by the data subject himself, the controller or an independent third party, a data trustee. An assignment rule must be created, e.g. through a reference table. The pseudonymization must be performed without the knowledge of the data subject. The data subject must be informed about the pseudonymization and it must be clarified who generates the pseudonym, who owns the assignment rule and under what circumstances an identification may take place. This is because pseudonymized data continues to be personal data. Particularly in the area of monitoring software development compliance with the GDPR is mandatory because it extracts its results from a data stream that contains personal data. The pseudonymization protects the data of the data subjects. At the same time, it is an opportunity to still being able to detect security incidents effectively and identify the responsible user in this regard.

3 Problem Statement and Research Questions

In SIEM systems very large amounts of data are processed from various sources, while at the same time a GDPR compliant data protection must be guaranteed. This can be achieved by protecting all data relevant to data protection against unauthorized access using techniques such as encryption or pseudonymization. Working on protected data, however, brings different additional problems with it. On the one hand, it needs to be ensured that incident recognition is still possible despite the data protection. On the other hand it also needs to be possible to remove the protection in case of an actual incident. These aspects, which we have identified as essential for a GDPR-compliant analysis process, translate into the following three specific research questions. In this work we use the pseudonymization for the realization of the data protection, since this represents a valid procedure according to both GDPR and different reporting regulatory environments.

3.1 Data Protection considerations and attacker model:

SIEM systems work with data from highly heterogeneous sources. As a result, different requirements need to be met in order to enable data protection in accordance with the GDPR. Establish data protection through the full encryption of all data would be the most intuitive and legally compliant way to process the data. However, since the GDPR only requires the protection of personal data, the data can also be classified according to protection requirements and partially peudonymized in this context. In this way it can be achieved to still be legally compliant, while more meaningful data is available for analysis at the same time. To achieve this, all acquired data needs to be available in a standardized form to allow the identification of information that needs to be protected. More specifically, the data acquired can be differentiated into information that is not relevant for data protection, data that may be relevant for data protection (e.g. path information in folder structures) and specific information relevant for data protection (e.g. e-mail addresses or contents of e-mails). In addition to this, the pseudonymization mechanism also needs to be protected. It must be ensured in a technical and organizational way for each data processing step within the SIEM system.

For being able to design a compliant and secure system, it is conducive, to define an attacker model, that determines the necessary measures. Thereby, the role, the goal, behaviour and the resources of the attacker are delimited:

- **Role:** The attackers role against which we consider our system protected can either be an outsider or an insider. An outsider is any person who has only access to interfaces of the system, which are open to the public. The outsider can however utilize a breach to gain access to certain parts or data of the system. Any third party who is involved in the SIEM system can also be referred to as an outsider. In contrast, an insider is any person who is directly involved into the system, such as analysts or server-admins.
- **Goal and behaviour:** The attacker can be either passive or active. The passive attacker only lists to the data without any intervention, whereas the active attacker tries to gain access to the data or the system by actively interacting with the system. For our approach, the considered goal of the attacker is to gain access to private data, since we design a SIEM system which is GDPR compliant.
- **Resources:** Since we utilize measures which are based on common asymmetric or symmetric cryptography, we can only consider attackers, with limited resources.

In summary, this raises the first question:

Q1: How must data that is processed in SIEM systems be protected to be GDPR compliant?

3.2 Impairment of Incident Identification through Data Concealment:

The GDPR stipulates that data protection must be applied as early as possible within the analysis process. Considering the data management of SIEM systems, this translates into a data protection obligation at the time of data acquisition. As a result, incident detection always needs to be performed on pseudonymization data. In this context, the relationship between pseudonymized data and plain text data within the data stream is a significant factor influencing possible analyses. This may impair both automated and manual analyses due to possible losses in the meaningfulness of the data analyzed. This leads to the second question:

Q2: Does the recognition of security incidents function properly despite of data pseudonymization or may losses and trade-offs be expected here?

3.3 Lifting the Pseudonymization while retaining Data Authenticity:

To enable the utilization of information about detected security incidents, while being compliant with the legislation, two main conditions need to be met. On the one hand, information about incidents must be available in the long term in an integrity preserving manner. On the other hand, a de-pseudonymization of the data needs to be possible at any time after detection. This warrants that the information can be used as a reliable means of evidence in trials that might take place in the future. Please note that the GDPR proposes both anonymization and pseudonymization techniques as possible data protection measures. However, since the use of anonymization would prevent the data from being used as evidence, this technique will not be considered further in this paper.

To achieve this, appropriate technical and organizational measures need to be in place ensuring that de-pseudonymization is only possible in case of actual incidents. Additionally, legal compliance also needs to be ensured for the data after lifting the pseudonymization for further processing. This requires protecting the data's integrity including origin authenticity.

Thus, the principle of data minimization complicates especially the goal of integrity. This principle has always been at the center of data protection and can be found in European and member state legal texts, e.g. already in the former Directive 95/46/EC and thus also in the GDPR. In detail Art. 5 GDPR describes that personal data shall be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed.

In general, there are two ways to conform with data minimization: (a) not collect or not forward personal data if it unnecessary, e.g. deleting it from data sets or blacken it out, i.e. redact it, or (b) making it harder to restore the personal data. For the latter, an important measure in this regard is the pseudonymization of data because it reduces the risks for the data subject and simultaneously it helps the controller to fulfil his data protection obligations (see also recital 28). By, for example, ensuring that only pseudonymized data is used during the incident analysis and that the person is only revealed if an anomaly is detected, would make a legally compliant processing of personal data in a SIEM conceivable. For the former, personal data, e.g. fields that contain this information, could just be removed before forwarding them.

On the other hand, both mechanisms for data minimization (pseudonymization or removal) result in a modification of the initially gathered data; an intended modification but a modification nevertheless. This results in standard cryptographic mechanisms to protect the data's integrity, such as digital signatures or message authentication codes, to fail. Thus, they are unsuitable to protect the integrity end-to-end. The more recent cryptographic mechanisms, known as redactable [18,41] or sanitizable signatures [1] are capable of allowing our architecture to authorize modifications such as removal of unnecessary data points from authentic data set gathered by the SIEM. From their initial versions these algorithms have evolved (see [2] for an overview). Most recently they undergo the process of becoming an internationally recognized signature standard[7]. While this process takes time, the current status shows that the cryptographic mechanisms have the needed maturity to be backed and accepted in the cryptographic community. Once becoming recognised through such a standard, legal argumentation for compliance becomes a lot easier as legislators and judges will find the algorithms in lists of known mechanisms. Even if not standardised (or not yet) the provided authenticity offerings are technically equivalent to normal signatures [32,16] and in any case much better than having none and also non-standard algorithms are suitable to win legal arguments in court cases – bearing the need for technical expertise appointed by court. This leads to the third question:

Q3: Which conditions need to be met to ensure that incident information can be de-pseudonymized in case of an incident and how can it be used as means of evidence?

4 Conceptualizing a GDPR-compliant SIEM System

Although SIEM systems have grown to mature security tools, privacy has largely been neglected in this area. Thus, we have previously defined central research questions that arise when applying the GDPR regulation. To answer these questions, we propose a SIEM architecture that is compliant with GDPR, while largely preserving its functionalities in this section. Therefore, we propose concrete solutions for each of the individual research questions based on an extended, GDPR-compliant architecture.

[7] ISO/IEC 23264 Redaction of Authentic Data https://www.iso.org/standard/78341.html [last accessed: Jan. 2020]

4.1 DINGfest base Architecture

This section gives an overview on our general security monitoring architecture and assigns the previously defined research questions to the respective areas of the architecture. The presented architecture is based on the general DINGfest architecture as presented in [23] and extends it by data protection measures and the resulting GDPR compliance. DINGfest is a research project that aims at improving the detection, forensic analysis and the reporting of detected incidents. The project started June 2016 and will finalize at the end of 2019 and is funded by the German Federal Ministry of Education and Research. The general system monitoring architecture is illustrated in Fig. 1. It consists of three main modules – namely data acquisition, data analysis and incident reporting located within an organization and shows an external authority possible counterpart for the receipt of detected incidents. The counterpart is intentionally included in the architecture design, since its role and the management of the data flows generated there are one central factor in ensuring the legal compliance of the system. This concerns data protection requirements according to the GDPR on the one hand and may also concern existing statutory reporting obligations of the organization.

The **data acquisition** module collects data from all monitored computing resources in the company. This data may contain personal data of employees and customers that needs to be protected. The monitored resources are not only computing devices like workstations, servers and mobile devices, but also network devices like routers and switches. The actual data is obtained from various sources. This includes, for example, data extracted with the help of Virtual Machine Introspection (VMI). This also includes data obtained from system log files or incident information provided by human sensors [46]. Moreover, all data extractions within the acquisition area are stored in the acquisition log. This enables a later auditability of all the information obtained. The extracted data is finally pushed into a larger data stream, which serves as data basis for the data analysis section. Data acquisition is the starting point at which all data (including personal data) is transferred to the system. As a result, research question Q1 must be addressed within this module to show how data must be protected or pseudonymized to ensure GDPR-compliant data handling. This additionally generates the required prudential value for the gathered evidence.

The **data analysis** module analyzes the whole data stream and tries to detect security incidents using a combination of fingerprinting and pattern recognition. If the detection engine discovers a potential security violation it generates an incident alert that contains a description of the assumed violation and the related data records. The alert is then received and analyzed by a forensic analyst. The analyst can use a visual analysis interface and request additional data from the data acquisition module. Should the suspicion be confirmed, the incident is forwarded to the reporting module. Otherwise, the incident alert is deleted right away. As shown above, the data acquired during data acquisition must be protected. This makes data analysis more difficult, since information is lost as a result of pseudonymization. Therefore, the research question Q2 will be ad-

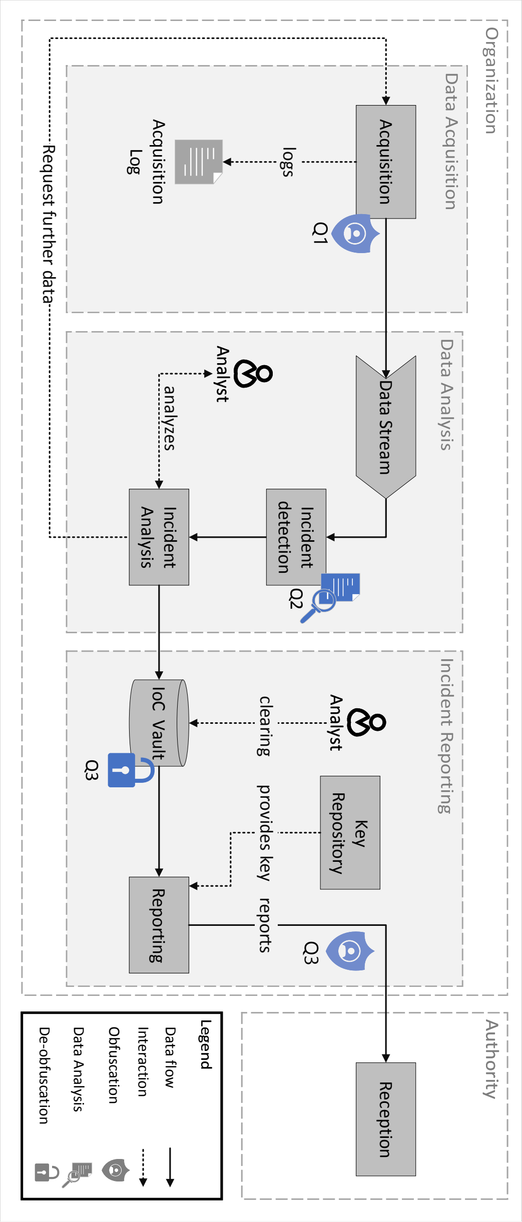


Fig. 1. DINGfest Base Architecture

dressed within this module to show how far incident detection with disguised data is still possible.

The tasks of the **reporting module** include the long-term storage of analyzed incidents (usually between several weeks and several year depending on the local legislation) and the reporting to local authorities in accordance with the legislation in force. Arriving incidents are therefore stored in a database and processed by an incident reporter. During this process the reporter might query the database to contrast the current incident with past incidents. Eventually a report is generated and forwarded to local authorities, in order to inform them or comply with regional regulations. Such reports may contain information about innocent individuals or company assets which also need to be protected. Within this module the research question Q3 will be addressed. The aim is to ensure that information can be de-pseudonymized in the event of an actual incident, while preserving its integrity. This is necessary to enable the use of the data as evidence in possible later court cases. Furthermore, it must also be ensured that the data can be reported to the appropriate authorities in compliance with the law and data protection regulations.

4.2 GDPR compliant Data Processing

In the following we present our approach in more detail, especially relevant parts, which enable GDPR compliant data processing inside SIEM (Q1). To this end, we propose an approach that pseudonymizes personal data at relevant points and at the same time allows to de-pseudonymize this data in case of a detected incident (Q3) in compliance with GDPR regulations. Fig. 2 shows the basic structure of this approach. To achieve a pseudonymization of the information, cryptographic methods are used. These on the one hand prevent access to personal data by encrypting it and on the other hand allow decrypting it under certain constraints specified by the GDPR. However, the decisive question is where the data must be encrypted and how to implement the key management for encryption.

The GDPR demands the protection of personal data as it is processed. Thus, we argue that personal data must be pseudonymized as soon as possible in the system. In the case of SIEM this is the case directly after or ideally during the data is acquired. To achieve this, a public key (B) is provided by a *TTP (Trusted Third Party)*, which is responsible for *key management*. With this key, the fields containing personal data are encrypted asymmetrically and the unencrypted personal data is deleted. In order to be able to comprehend and proof, that all personal data has been pseudonymized, an *Acquisition Log* is kept. For this purpose, we propose to use a tamper proof logging scheme, which synchronizes all logged data with an external blockchain as presented by Putz et al. [35]. For proper use of public key cryptography, we refer to López et al. [22].

In order to determine, which fields must be encrypted, a *Policy* is followed. For each logging system type an individual mapping must be defined that specifies the fields containing personal data. This policy is defined by the *Data Protection Officer (DPO)* or an equivalent position inside the organization. A DPO

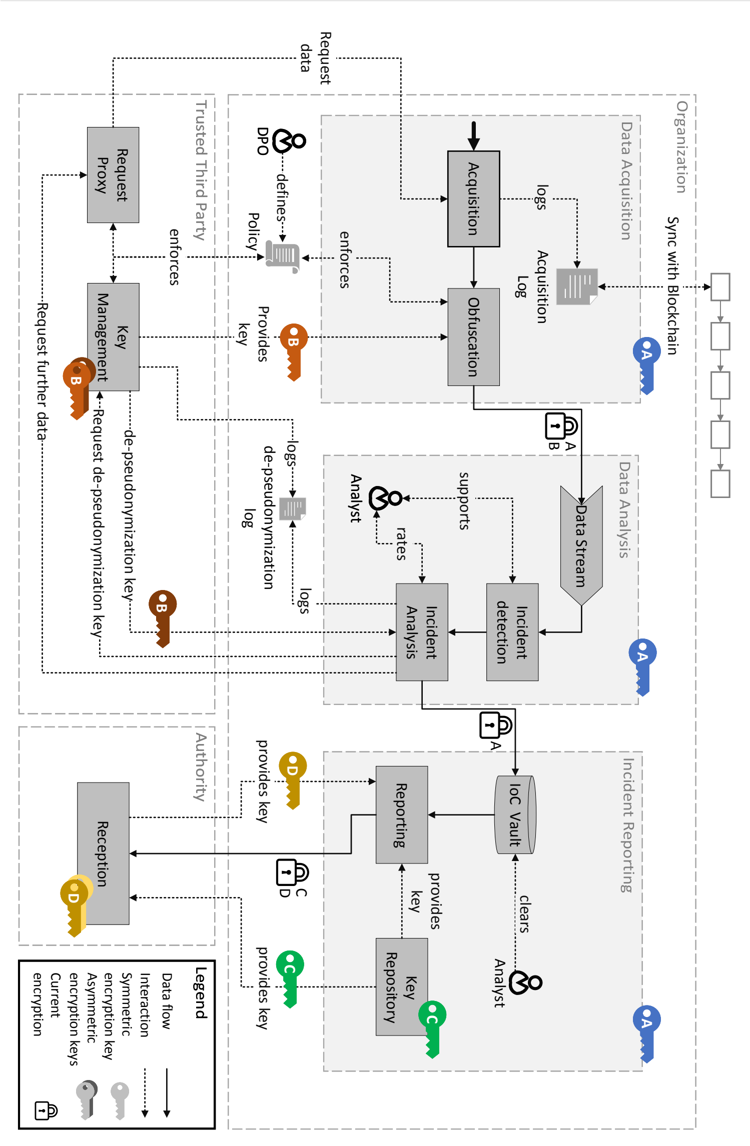


Fig. 2. DINGfest GDPR Architecture

is responsible for compliance and data protection within an organization and should have the necessary expertise to make the required decisions.

In addition to the data protection aspects shown above, the model presented is also intended to provide protection against attacks resulting from the attacker model defined in Section 3. According to the assumptions made, possible dangers from insiders and outsiders are examined more detailed in the following.

Outsider: According to the model, outsiders can be divided into two main groups. Common outsiders, which have no specific reference to the system, and the TTP as an outsider, which is partially involved in the analysis process. The authority as the third participant is not considered in detail here, as it is supposed to have access to the data it receives in the context of a report.

- **Common outsider:** The major problem is to prevent data flows to outsiders. Specifically, the areas of data acquisition, data analysis and incident reporting must be protected. This is essentially guaranteed by a consistent use of the internal, symmetric key A. This ensures that the data remains protected even in the event of an unwanted extraction. The data may only be possibly unprotected in the case of an extraction during the data acquisition process. The security of this data mainly depends on the level of protection of the underlying source system.
- **TTP:** The TTP used in the present data model also represents a specific outsider, who is integrated into the SIEM process. However, the TTP only receives the key B from the data flow for custody, but does not have access to data from the data stream at any time. It is also worth noting that the TTP does not have access to the key A at any time. Accordingly, the TTP must be considered the equivalent of other outsiders in the case of data leaks.

During data analysis, an incident detection approach is followed. This approach is mainly automated but can also be supported by human analysts. Thereby, the incident detection is conducted solely on pseudonymized data and thus is GDPR compliant. The thereby used event detection approach is elaborated further in the following chapter.

Insider: In the present model, only two groups of people have access to the internal data. These are analysts in the areas of data analysis and incident reporting on the one hand and data protection officers who define the corresponding policies on the other hand.

- **Analyst:** The analysts involved are only provided with specific data extracts and personal data under certain circumstances. For this purpose, an approval for specific data components must be granted according to the policy defined by the data protection officer. If such an approval does not exist, analysts always work only with pseudonymized data.
- **DPO:** The data protection officer is never given access to the data within the data stream and thus has access to resources that are equivalent to an outsider. On the other hand, the DPO has a protective influence on the data

stream by defining the respective policy. This ensures that the protection of the data stream is always split between two different roles within the company.

4.3 Event Detection on protected Data

Different software usually comes with different log formats that is often loose text. In our case, we use standard Linux logs like *syslog* and *auth.log* that usually come with a Linux distribution. Furthermore, we use *access.log* of Apache's HTTP Server [42]. Since system call traces are a rich source of behavioral information [36,19], we also use system calls traces that are obtained via virtual machine introspection. System call tracing has a negative impact on performance, but especially enterprise environments can benefit since some events cannot be detected using common logs.

Table 1. The unified log message format [21]

Name	*Description*
source	The source from where the message comes from.
type_id	Describes the type of the message, e.g., the system call number.
date	Timestamp of the message when it was generated.
path	A path, e.g., which path was opened.
user	The user who performs the event.
process_name	The name of the process that performs the event.
...	...
misc	Can be used for random things (no personal data) that do not fit into that format.

Log messages are transformed into a unified structured log format. An excerpt of the message format that we use in DINGfest can be seen in Table 1 [21]: The entry *source* specifies from which of our sources log the message comes from. Thereby, we assume that it is not possible to deduce from the source to the user, i.e., in server scenarios. One of the most important attributes of the unified log message is *type_id*. This ID specifies what kind of message it is. In case of system calls the *type_id* is the system call number. *Misc* may contain arbitrary information that does not fit into the unified message format, e.g. command line option. We assume, that this field does not contain personal data. For the evaluation we checked manually that this field does not contain personal data. Another useful feature is *path*. However, the path may contain personal data such as the user name.

An example of a unified log message is shown in Listing 1.1

```
{
    ”source”        : ”syscalls”,
    "type_id"        :  59,
    "process_name"   :  "ls",
    "user"           :  "alice",
    "pid"            :  103,
    "path"           :  "/home/alice/topsecret/"
    ...
}
```

Listing 1.1. Example of a unified message.

So we can distinguish between three kinds of log file entries:

1. Those that definitely contain personal data (e.g., user),
2. those that may contain personal data (e.g., path), and
3. those that do definitely not contain personal data (e.g., source, type_id, process_name, misc)

The classification may vary from system to system. For example, it is also possible that in a specific scenario a process name or a source name may also contain personal data. The classification, however, determines which features can be used for privacy friendly event detection, namely only features from the third category. We use this idea in our evaluation in Section 5.1 to assess the impact of privacy protection on event detection quality.

4.4 De-pseudonymization in case of an incident

In the previous section we presented an approach that allows us to perform incident detection on data that is pseudonymized according to the GDPR regulation. On this basis, this section describes how security incidents can be de-pseudonymized after detection in order to analyze and process them further and to prepare them for a legally compliant report. The complete process for these descriptions is additionally shown in Fig. 3.

When the automated data analysis found indications for a possible incident, it is first necessary to verify the result. For this purpose, the data is revised by an analyst to ensure that it is an actual incident and to avoid false positives. Once the analyst approves the incident within the *Data Analysis* module, the *Trusted Third Party (TTP)* is contacted, which initially provided the public key B for the data pseudonymization within the *Data Acquisition* module. The *TTP* receives the signature of the data packet concerned in order to be able to identify the appropriate key and checks the request against the decryption policy specified by the DPO. If the check is negative, the request is denied. If the check is positive, the *TTP* determines the correct private key B for the signature provided and sends it to the *Data Analysis* module to allow the de-pseudonymization of the data. When de-pseudonymizing incident data, it is also important to enable auditing of de-pseudonymizations. Therefore, we store every de-pseudonymization request and sign it, in order to be able to provide proof of data access afterwards.

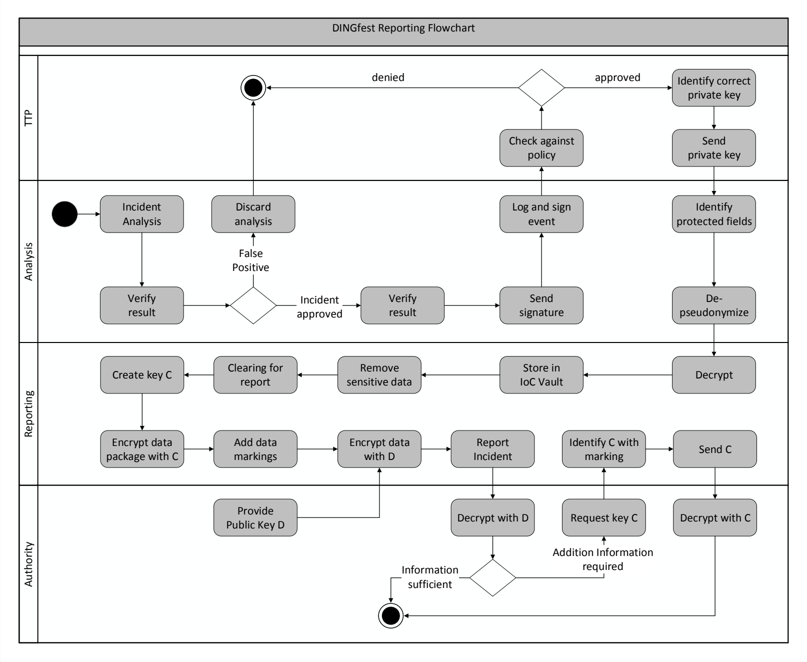


Fig. 3. DINGfest Reporting Flowchart

After the analysis module has received the private key B from the *TTP*, the initially appended data markings are used to identify all fields that contain pseudonymized information within the data package and to de-pseudonymize it. The resulting data package is then transferred to the Incident Reporting module where it is decrypted using the key A and stored within the *IoC Vault*, which is an integrity proof, long term storage for incident data as shown by Boehm et al. [4]. This data can then be used for further analyses and incident reports. If an incident must be reported, the data needs to be cleared by an analyst first. This is important to prevent both privacy violations of personal data and the publication of confidential company data. More specifically, the analyst must decide, which data is to be excluded from the report and which is to be secured. This additional information about performed pseudonymizations and exclusions is appended to the data using further data marking definitions. The actual extent of data protection, however, may strongly depend on the use-case of the report. While in the case of reports within the scope of a reporting obligation, the statutory requirements must be complied with, in the case of voluntary reports significantly more data can be concealed or removed.

After an analyst has cleared an incident and prepared it for a report within the *IoC Vault*, it is transferred to the *Reporting* module in the next step. A symmetrical key C is created in the key repository and assigned to this very data package to pseudonymize the previously chosen contents. Furthermore, the package is extended by an additional data marking that contains the signature of the utilized key C key to enable later attributions. The key C is then used to pseudonymize the data according to the analyst's specifications. In addition to this, the data is also encrypted with the public key D, provided by the recipient (for example, an authority). This ensures that the reported data can only be opened by the correct recipient, i.e., a legal authority. In a final step, the data (secured with keys C and D) is transferred to the recipient. If the data was reported due to a reporting obligation, such as European NIST directive [11] or German IT-security law [9], the receiving authority may request a decryption afterwards under certain conditions. In this case, the authority needs to be able to request the key C from the organization. In order to receive the key C, the authority transmits the key signature contained in the data package to the organization. This enables the assignment of the correct key. If the correct key C is assigned, it can be transferred to the authority.

4.5 Evidence generation using malleable signatures which withstands pseudonymizations

We can positively answer the second half of Q3, i.e. we can ensure that we are able to use the pseudonymized or partial reported events as means of evidence. Assume we add protection of integrity and origin authentication during the data gathering inside the data acquisition module. Inside DINGfest's base architecture, the evidence could be protected by standard electronic signature schemes, e.g. the different acquisition modules would sign the data they gathered. When a cryptographic signature algorithm complies with the requirements of common legal frameworks for electronic signatures its signature provides a high probative value for the data being signed. This said, any subsequent modification for data-protection compliance would destroy any evidence guarantees for the remaining data, e.g. removing the full path from a signed full file name of a malicious executable as it contains a personally-identifiable user name also removes the evidence protection for the name of the executable. Thus, we propose use redactable or sanitizable (malleable) signature schemes to retain the authenticity and integrity of the data gathered from the data acquisition module towards the final report. This means that, if wanted, the digital signature protects the authenticity of the data provided even till the incident report, i.e. so in the final report one can verify that the event data has not been modified in unauthorized ways –not tampered with– and that it originated from a trusted data acquisition module.

By omitting the cryptographic details of other malleable signature schemes, the privacy statement describes, which parts of the data could be removed or pseudonymized. This allows to control the signature scheme for which subsequent changes are made and parts are authorized. Thus, removing the original data or encrypting these parts, would allow the subsequent steps to always verify the authenticity of the remaining data. When data is de-pseudonymized the best-suitable malleable signature schemes are those that offer mergeability [33]. This allows to put data parts back into the signed data set and thus the original malleable signature would now verify over all remaining parts, the ones previously readable plus those added by the de-pseudonymization. If not added, private malleable signature schemes [2] retain the confidentiality of all those parts that have been removed, i.e. even though one can successfully verify the signature on partial data, the information contained in the signature itself does not allow an attacker to gain information on the data parts removed and thus not shared. Hence, the added value of a retain-able private malleable signature, like [33], does not violate any GDPR requirements [30].

The legal analysis of these private accountable redactable signature scheme shows that they increase the legal probative value for the signed reported data as eIDAS compliant electronic signatures could provide [30,16,31,45]. Hence, the DINGfest GDPR architecture protects the records such that the remaining information can be used as means of evidence; further after the de-pseudonomization steps at any later time the data's origin and originality is attested.

5 Evaluation

In the previous sections we presented an architecture for a GDPR compliant SIEM system. The central elements of our approach are to guarantee a GDPR compliant data processing, to enable the recognition of security incidents on pseudonymized data as well as the de-pseudonymization of the data in the case of an incident. In this section we evaluate the validity of our approach in two ways. First, we conduct a technical evaluation of the impact of pseudonymization on the detectability of events. Subsequently, we carry out a legal evaluation for our proposed solution. To achieve this, we investigate the individual components of our architecture presented in Section 4.2 on conformity with the specifications of the GDPR as shown in Section 2.2.

5.1 Impact on Detectability

Evaluation Methodology The evaluation of the impact of privacy protection on the quality of SIEM is performed based on the theory for forensic fingerprint calculation of Dewald [10].

In this theory, all interactions of interest with the system (e.g., by users) are called *events*. An example event is the login of a user. Many events either directly or indirectly leave digital traces within the system (e.g., in log files on disc or in main memory). These traces are formalized as *feature vectors*. Generally, a feature is a quantifiable attribute of a system that can be observed by the SIEM. In our study we concentrate on feature vectors that can be extracted from log files system call traces [21]. Obviously, tracing all system calls is very expensive in terms of performance, there are ways to get rid of most overhead caused by system call tracing. The theory [10], which we now explain, defines conditions under which an event is detectable based on the features traces it leaves in the log files of a system.

The set of features that we consider in our evaluation is based on an abstract representation of log file entries and attempts to harmonize many log files in modern systems. Our format represents every log file entry using the following four *features*:

- a *source* from what log the message comes from,
- a generic *type_id* that describes the kind of log message,
- a *path*, and
- a *misc* field that may contain arbitrary content (e.g., the name of a network adapter).

A *feature vector* is a vector of values for these features. Depending on the system, there can be many different features vectors consisting of these four features. Since an event can cause multiple entries in multiple log files, we define the set of feature vectors that are generated as the *evidence set* of that event.

More formally, let Σ be the set of all possible events that can happen in the system and are of interest to the SIEM. When some event $\sigma \in \Sigma$ happens, log entries are generated. The *evidence set* $E(\sigma)$ of event σ is the set of all subsets of feature vectors that are thereby generated by σ. It is technically necessary, that the evidence set is closed under subsets. Intuitively, it can be interpreted as the fact that partial evidence is also evidence of the event.

It is obvious that the evidence sets of different events may overlap. To be able to detect an event, it is necessary to calculate the *characteristic evidence set* $CE(\sigma)$[10] of an event σ, which is defined as the set that contains only feature vectors that are caused by σ and *not* by any other event $\sigma' \in \Sigma$. Formally, the set of characteristic evidence of an event σ with respect to a set of other events Σ' is defined as follows:

$$CE(\sigma, \Sigma') = E(\sigma) \setminus \bigcup_{\sigma' \in \Sigma'} E(\sigma')$$

The set of characteristic evidence of an event is also called *characteristic fingerprint* of that event.

As one can see in the formula above, a characteristic fingerprint is defined for a specific reference set Σ'. All feature vectors that are caused by $\sigma' \in \Sigma'$ are not in $CE(\sigma, \Sigma')$. One can say, that $CE(\sigma, \Sigma')$ is the evidence set $E(\sigma)$ minus all other evidence sets of events in Σ'. Let $|CE(\sigma, \Sigma')|$ and $|\Sigma'|$ be sufficiently large, then a match of the feature vector with the log files of a system is a clear indication that σ happened and not σ'. The size of CE is an indication for the discriminative power of the evidence. The larger the set, the higher is the probability that the event may be detected no matter what the reference set Σ' looks like. However, it is also possible, that $CE(\sigma, \Sigma')$ is empty, i.e., one cannot detect reliably the occurrence of σ.

Characteristic Evidence without Personal Data We now evaluate the impact of pseudonymization on the existence of characteristic evidence that is needed for event detection. The evaluation setting is based on the DINGfest architecture as described by Latzo and Freiling [21] [20]. We calculated evidence and characteristic evidence sets for 45 different events (see also Table 2 that typically appear in Linux server environments as one typically finds them in small and medium-sized enterprises. In our threat model, we consider an adversary with root privileges that were either gained via a privilege escalation attack or by having them anyway (i.e., a malicious insider). Basically, it is not possible to determine the intention of an administrator's input. Hence, most events can also be used maliciously, e.g., for information retrieval, covering traces, etc. So, basically all events might be interesting during a forensic analysis or event detection.

The higher the number of feature vectors in a characteristic fingerprint, the better the quality of that fingerprint. This is intuitive since a feature vector in a fingerprint is basically an indicator of an event. In the evaluation, we compare the size of characteristic evidence sets with and without taking personal data into account. More concretely, we consider the following two feature sets:

- $F_1 = \{source,\ type\_id,\ path,\ misc\}$
- $F_2 = \{source,\ type\_id,\ misc\}$

First, F_1 is a feature set that has turned out to be reasonable for our events. However, F_1 includes the *path* feature that may contain personal data. Features of F_2 do not contain personal data. For calculating the fingerprints, an event was executed 40 times (trainings set). Furthermore, the reference set Σ' for a characteristic fingerprint of σ are always all other events. Table 2 compares sizes of the characteristic evidence set using the two feature sets including the decrease rate when using F_2 instead of F_1. As one can see, omitting the path as a feature has a huge impact on the size of characteristic evidence. On the average, using F_2 reduces a characteristic fingerprint by half of its feature vectors. For three events, there is no characteristic fingerprint, anymore. Figure 4 shows the absolute number of feature vectores of the characteristic evidence sets using F_1 and F_2. "Big"events, that come originally with big characteristic fingerprints are in general more affected than smaller events.

We have shown that it is possible to calculate characteristic evidence for all events even if features that contain personal data are not used. However, the fingerprints that we generated had a lower quality, i.e., the size of the characteristic evidence set was reduced by an average of about 50%. By extending the feature set and the set of traces acquired from the SIEM, we conjecture that fingerprints can also be calculated for this action even if data is pseudonymized.

In the following we want to compare the matching results using characteristic fingerprints with F_1 and F_2. For matching, we calculate a score that indicates what proportion of feature vectors in event traces are matched by a characteristic fingerprint. Fig. 5 (F_1) and Fig. 6 (F_2) show the corresponding matching matrices. The values there are average values of 10 traces of the event (test set). It stands out that the matching matrices are quite similar. In Fig. 6 there

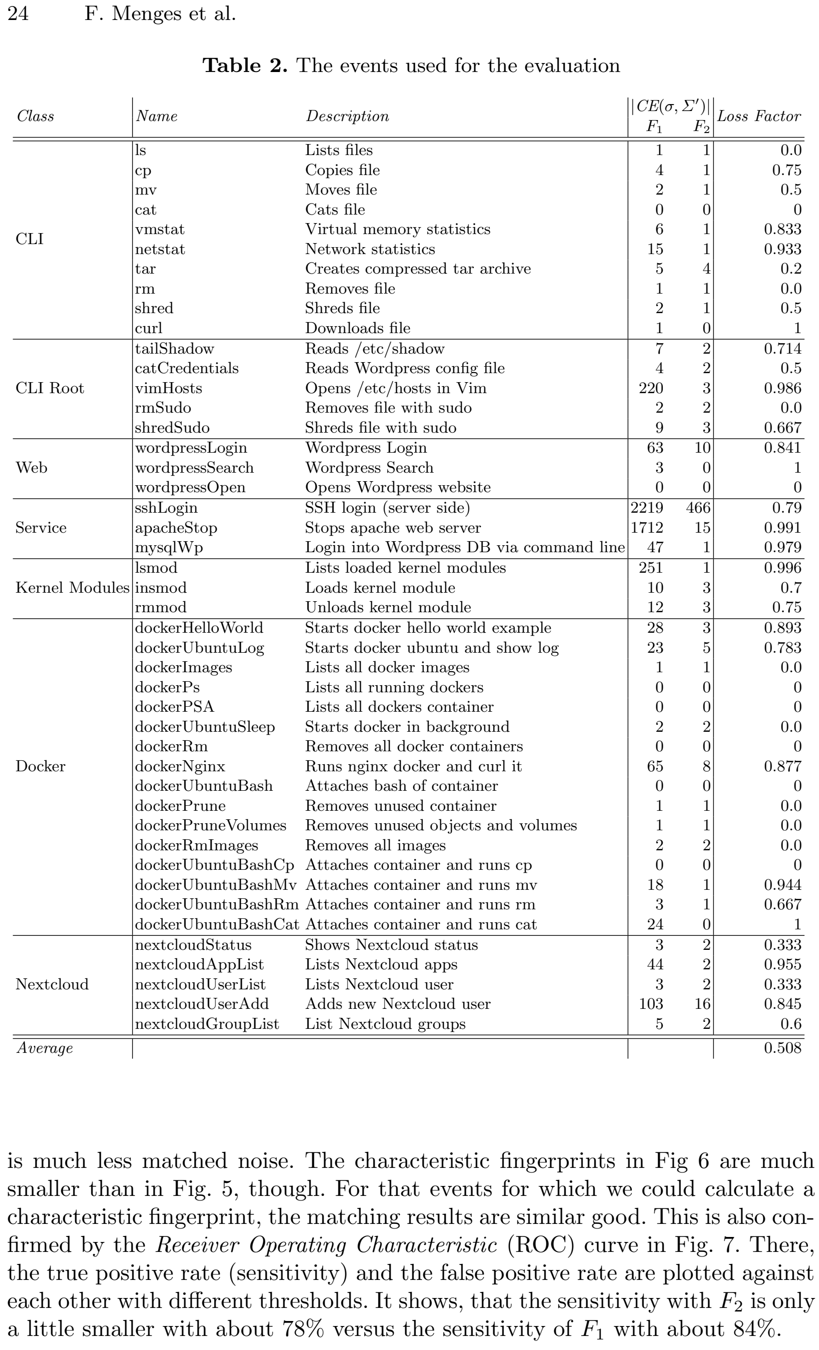


Table 2. The events used for the evaluation

Class	*Name*	*Description*	$\lvert CE(\sigma, \Sigma')\rvert$		*Loss Factor*
			F_1	F_2	
CLI	ls	Lists files	1	1	0.0
	cp	Copies file	4	1	0.75
	mv	Moves file	2	1	0.5
	cat	Cats file	0	0	0
	vmstat	Virtual memory statistics	6	1	0.833
	netstat	Network statistics	15	1	0.933
	tar	Creates compressed tar archive	5	4	0.2
	rm	Removes file	1	1	0.0
	shred	Shreds file	2	1	0.5
	curl	Downloads file	1	0	1
CLI Root	tailShadow	Reads /etc/shadow	7	2	0.714
	catCredentials	Reads Wordpress config file	4	2	0.5
	vimHosts	Opens /etc/hosts in Vim	220	3	0.986
	rmSudo	Removes file with sudo	2	2	0.0
	shredSudo	Shreds file with sudo	9	3	0.667
Web	wordpressLogin	Wordpress Login	63	10	0.841
	wordpressSearch	Wordpress Search	3	0	1
	wordpressOpen	Opens Wordpress website	0	0	0
Service	sshLogin	SSH login (server side)	2219	466	0.79
	apacheStop	Stops apache web server	1712	15	0.991
	mysqlWp	Login into Wordpress DB via command line	47	1	0.979
Kernel Modules	lsmod	Lists loaded kernel modules	251	1	0.996
	insmod	Loads kernel module	10	3	0.7
	rmmod	Unloads kernel module	12	3	0.75
Docker	dockerHelloWorld	Starts docker hello world example	28	3	0.893
	dockerUbuntuLog	Starts docker ubuntu and show log	23	5	0.783
	dockerImages	Lists all docker images	1	1	0.0
	dockerPs	Lists all running dockers	0	0	0
	dockerPSA	Lists all dockers container	0	0	0
	dockerUbuntuSleep	Starts docker in background	2	2	0.0
	dockerRm	Removes all docker containers	0	0	0
	dockerNginx	Runs nginx docker and curl it	65	8	0.877
	dockerUbuntuBash	Attaches bash of container	0	0	0
	dockerPrune	Removes unused container	1	1	0.0
	dockerPruneVolumes	Removes unused objects and volumes	1	1	0.0
	dockerRmImages	Removes all images	2	2	0.0
	dockerUbuntuBashCp	Attaches container and runs cp	0	0	0
	dockerUbuntuBashMv	Attaches container and runs mv	18	1	0.944
	dockerUbuntuBashRm	Attaches container and runs rm	3	1	0.667
	dockerUbuntuBashCat	Attaches container and runs cat	24	0	1
Nextcloud	nextcloudStatus	Shows Nextcloud status	3	2	0.333
	nextcloudAppList	Lists Nextcloud apps	44	2	0.955
	nextcloudUserList	Lists Nextcloud user	3	2	0.333
	nextcloudUserAdd	Adds new Nextcloud user	103	16	0.845
	nextcloudGroupList	List Nextcloud groups	5	2	0.6
Average					0.508

is much less matched noise. The characteristic fingerprints in Fig 6 are much smaller than in Fig. 5, though. For that events for which we could calculate a characteristic fingerprint, the matching results are similar good. This is also confirmed by the *Receiver Operating Characteristic* (ROC) curve in Fig. 7. There, the true positive rate (sensitivity) and the false positive rate are plotted against each other with different thresholds. It shows, that the sensitivity with F_2 is only a little smaller with about 78% versus the sensitivity of F_1 with about 84%.

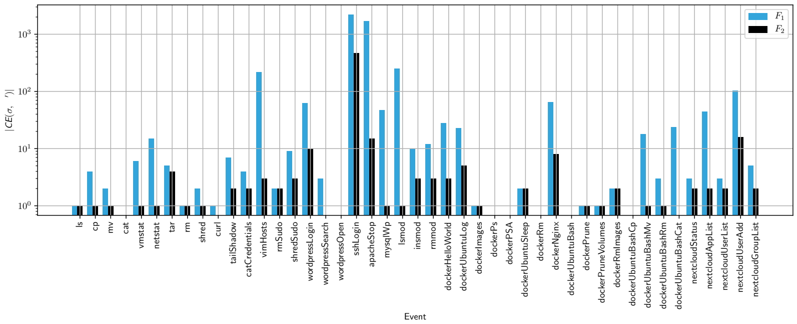


Fig. 4. Comparison of powers of characteristic evidence sets with personal data (blue) and without personal data (black).

In this section we showed that it is possible to calculate characteristic fingerprints with a reduced feature set that does not contain personal data. While the size of the characteristic fingerprints decreased when using that feature set, the matching results were very similar. It showed, that when there is a characteristic fingerprint, matching usually also works. So, to improve this approach, future work should focus in extracting more features from logs that help to increase the size of characteristic fingerprints.

5.2 Legal Evaluation

As we have seen above, to generate fingerprints of high quality, data must be obtained by processing previously collected data during the data acquisition which clearly relates to a personal user's actions and thus is considered as personal data. Hence, the general principles mentioned in Sec. 2.2 relating to processing of personal data and especially a lawful processing are important. Processing shall be lawful only if one of the Art. 6 GDPR included reasons applies. Applicable and best suited for the SIEM case is Art. 6 (1) lit. f, when processing is necessary for the purposes of the legitimate interests. This clause is different to the other lawful bases as it is not centered around a particular purpose and it is not necessary that the individual has specifically agreed to (consent). Legitimate interests are more flexible and could in principle apply to any type of processing for any reasonable purpose. Art. 6 (1) lit. f states:

”1.Processing shall be lawful only if and to the extent that at least one of the following applies: (f) processing is necessary for the purposes of the legitimate interests pursued by the controller or by a third party, except where such interests are overridden by the interests or fundamental rights and freedoms of the data subject which require protection of personal data, [...]”.

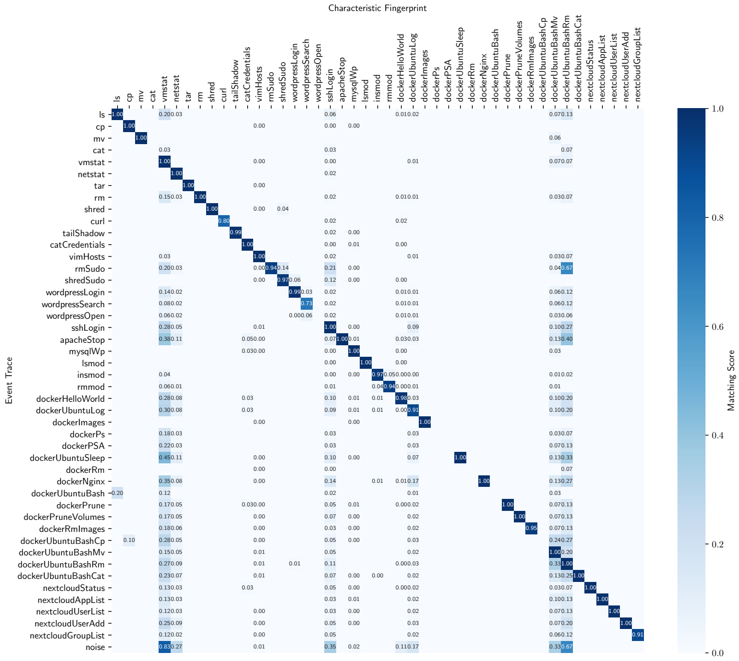


Fig. 5. Matching matrix using F_1. The events listed on the y-axis are the ground truth, the events on the x-axis correspond to the characteristic fingerprints [20]

Legitimate interest is balanced with personal data protection Since legitimate interests can apply in a wide range of circumstances, it is mandatory that the controlling party puts its legitimate interests and the necessity of processing the personal data to the interests, rights and freedoms of the individual in balance. To provide a balance-test, the key elements of the legitimate interests provision is contained in a so-called three-part test. Whereas this test is not explicitly named in the GDPR, the legitimate interests provision does incorporate three key elements:

- Purpose test: there must be a legitimate interest behind the processing.
- Necessity test: the processing must be necessary for that purpose.
- Balancing test: the legitimate interest must be balanced with the individuals interests, rights or freedoms.

This concept of a three-part test for legitimate interests has been confirmed by the Court of Justice of the European Union in the Rigas case (C-13/16, 4 May 2017) in the context of the Data Protection Directive 95/46/EC, which contained a very similar provision. This means, the controller must be able to

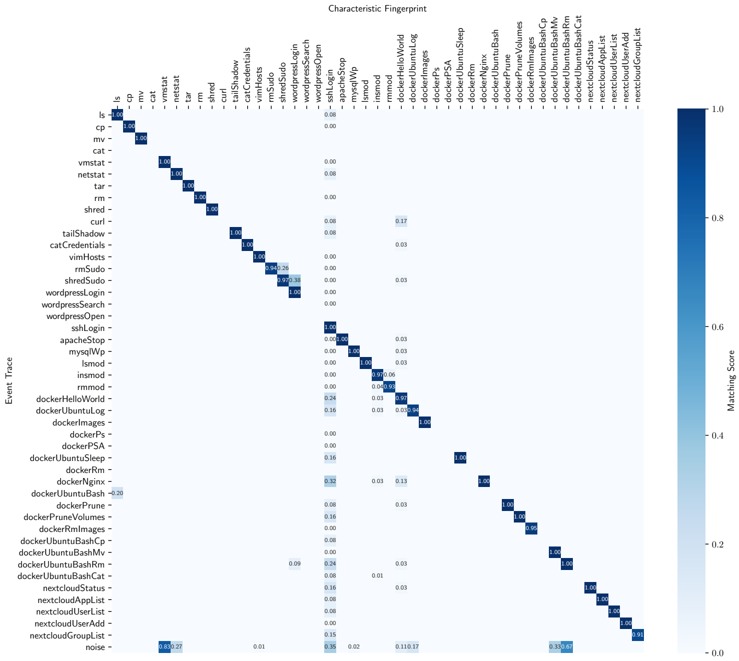


Fig. 6. Matching matrix using F_2. The events listed on the y-axis are the ground truth, the events on the x-axis correspond to the characteristic fingerprints.

meet all three requirements of the test prior to commencing the processing of personal data.

Firstly, **purpose** is clearly given as the whole purpose of the SIEM architecture as given in Sec. 4 is to detect unlawful use of information systems and their data, it is important to make clear that the European Parliament has already considered the legitimate interest of processing personal data necessary for the purposes of preventing fraud. This is explicitly backed by recital 47: "*[...] The processing of personal data strictly necessary for the purposes of preventing fraud also constitutes a legitimate interest of the data controller concerned. [...]*".

Secondly, with regards to the condition relating to the **necessity** of processing personal data, it is important that derogations and limitations in relation to the protection of personal data must apply only in so far as is strictly necessary. In that regard, communication of features which do not contain personal data, does not make it possible to identify a person with enough precision in order to be able to bring an action against him. Accordingly, for that purpose, it is necessary for the SIEM system to obtain also the possibility of full identification of that person, i.e. allow to de-anonymize and retain authenticity proofs in order

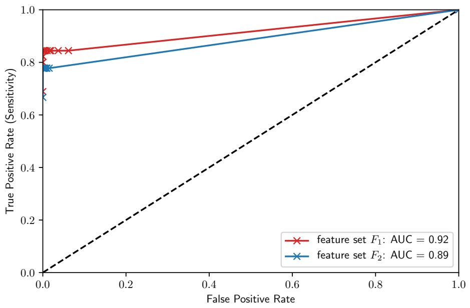


Fig. 7. ROC curve with F_1 in red and F_2 in blue. The differences are quite small.

to construct substantial and reliable evidence of an unlawful use of the system against that person.

Thirdly, it is necessary to make a **balancing test** to justify any impact on individuals. During the test the controller takes into account ”the interests or fundamental rights and freedoms of the data subject which require the protection of personal data”, and makes sure they dont override his interests. In recital 75 speaks of the risks of the rights and freedoms of natural personas: *”The risk to the rights and freedoms of natural persons, of varying likelihood and severity, may result from personal data processing which could lead to physical, material or non-material damage, in particular: where the processing may give rise to discrimination, identity theft or fraud, financial loss, damage to the reputation, loss of confidentiality of personal data protected by professional secrecy, unauthorized reversal of pseudonymisation, or any other significant economic or social disadvantage; where data subjects might be deprived of their rights and freedoms or prevented from exercising control over their personal data; [...]; where personal aspects are evaluated, in particular analyzing or predicting aspects concerning performance at work, [...] ; or where processing involves a large amount of personal data and affects a large number of data subjects. ”*

Since the data acquisition module collects data from all monitored computing resources in the company, one can assume a great danger for personal data of employees and customers. Also, the analysis of personal data in the data stream by fingerprinting and pattern recognition and especially the merging of data is in general - interfering with the privacy rights of a natural person. And finally, is the reporting module and the included long-term storage of analyzed incidents as well as the reporting to the authorities itself a potential risk for personal data. Since the complete monitoring of the users without cause is not compliant with the GDPR, especially since the user does not have any possibility to intervene, fundamental rights would be violated, if the controller is not implementing appropriate technical and organizational measures to ensure a level of security appropriate to the risk, including inter alia as appropriate, for example the pseudonymisation and encryption of personal data.

Mechanisms for GDPR-compliance in DINGfest architecture DINGfest's GDPR architecture counters the above-mentioned problems by implementing special steps as part of their work flow for a GDPR-compliant SIEM system:

First of all, by a continuous pseudonymisation through obfuscation during the data acquisition. The suspension will only be carried out under certain conditions determined by controller and, in particular, in case of suspicion of a criminal offence. Since the public key is always provided by a trusted third party (TTP) and policies provide the organizational background before a special field gets encrypted, the balance between the rights of the controller and the user should be met. All technical steps in which personal data is processed are accompanied by a special pseudonymization method through obfuscation. If data analysis has then found indications for an (possible) incident, the data protection officer has to approve this case as an "incident case" within the data analysis module. Only then the TTP receives a key identifier for the data packet - not the packet's contents, not even in pseudonymous form -, in order to find the appropriate key. The critical point is the policy (see the "Policy" defined by the Data protection office (DPO) in Fig. 2), which is being consulted by the TTP before sending the decryption key to the data analysis module. This organizational measure ensures a level of security appropriate to the risk, which is to reveal private data to the controller. By providing a log of every request to de-pseudonymize data fields the architecture enables to comply with transparency requirements, like the right of access.

Only after passing this safety measure the DINGfest architecture allows to reveal data to the controller (the data analyst) using the private key B, to de-pseudonymize all necessary fields that contain pseudonymized information within the data package. Only if the analyst decides to include this in the report the resulting data package is then transferred to the incident reporting module, during transfer and storage it gets again encrypted under key A. Again, the access to the encrypted long-term storage of the data within the IoC vault, including the use of data for further analysis and incident reports, only applies for cases that deserve an attention because of potential unlawful behavior. This is clearly a legitimate interest which is not overridden by recital 47 of the GDPR. Furthermore, it depends on the individual use case, which data is to be excluded from the report and what data must be removed or stays pseudonymized. This extra step, in which the analyst balances the rights of both parties, is the very essence of the balance test, and here the DINGfest GDPR architecture implements this check to balance the legitimate interest with the individuals interests. As said, if the data analysis module decides to keep pseudonymization of fields then only pseudonymized data of this incident is transferred to the incident reporting module. When the report is generated, the data is again encrypted with the public key of the recipient in most cases an authority. This way, the reported data can only be opened by the correct recipient.

DINGfest's GDPR architecture reduces the risks for personal data By using the methods of pseudonymisation and encryption of personal data, it is to be concluded that scope, context and purposes of processing as well as the risk of varying likelihood and severity for the rights and freedoms of natural persons, the controller has implemented appropriate technical and organizational measures to ensure a level of security appropriate to the risk.

6 Conclusion

In this paper we presented an architecture for a GDPR compliant SIEM system, as implemented in the DINGfest prototype SIEM system. We first identified central questions that must be answered for the development of such a platform. The questions affected the necessary conditions for a GDPR compliant data processing, techniques for the incident recognition on pseudonymized data as well as a lawful de-pseudonymization techniques in the case of occurred incidents. We then answered these questions with the help of our architectural design and evaluated them both from a technical and legal perspective. Using this evaluation, we have shown that it is possible to comply with the legal requirements for pseudonymization, while at the same time keeping detectability. Altogether we presented a base architecture for a GDPR compliant SIEM system with this work. Although it was developed based on our underlying system DINGfest, it may also serve as a draft for other security systems that have to be adapted to GDPR specifications.

In the context of this work, it was revealed that the performance of the recognition mechanisms used can be impaired using pseudonymization. This is one challenge that could be addressed in future work. Beyond that we defined the fundamental boundaries of a GDPR conform architecture with this work. However, various details were not considered. An example for this is to transfer our architecture to already established SIEM systems. Each system is tailored to its infrastructure and thus, it is necessary to define, which of the collected data sets needs to be protected. This applies both to data that is collected during initial data acquisition and to data that is prepared for a report. To support this process, it would be helpful to develop a central repository that defines the data points relevant to data protection for frequently used data sources. Furthermore, it will also be necessary to develop the needed details for the data protection policy within SIEM systems in future works. It would be conceivable to develop a generally applicable basic policy and specific implementations of this policy adapted to individual systems.

Regarding the legal probative value DINGfest using malleable signatures allows to balance integrity protection for evidence and GDPR-compliant removal or pseudonymization of the gathered data. To achieve this the data acquisition module emits malleable signed data –instead of simply digitally signed data– and hence any subsequent modification due to GDPR-compliant processing does not inhibit the verification of the integrity and origin of the remaining data. With a scheme that is accountable and private and supports mergeability, previously obfuscated parts of an entry can be subsequently de-obfuscated and the signature still verifies and provide means of evidence.

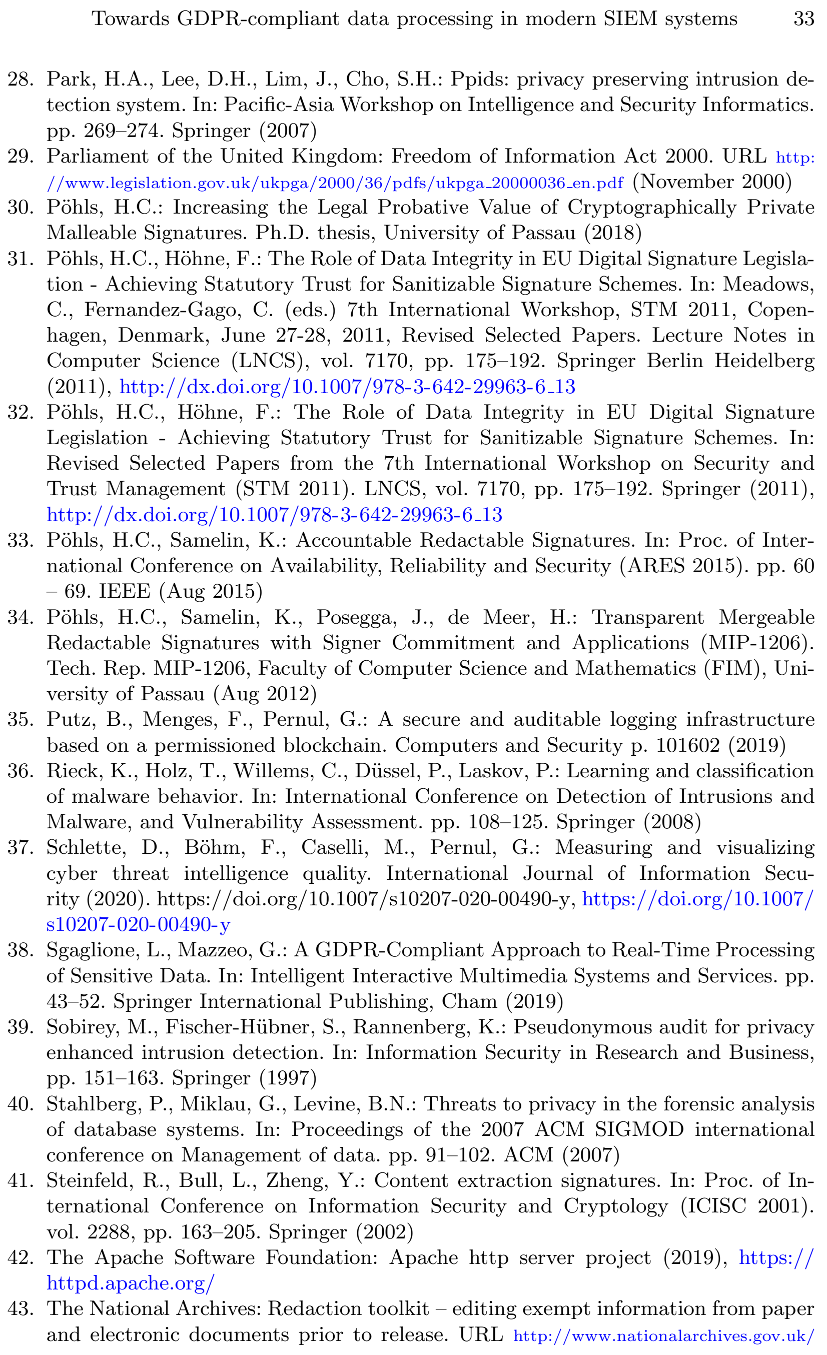
Acknowledgement

This research was supported by the Federal Ministry of Education and Research, Germany, as part of the BMBF DINGfest project (https://dingfest.ur.de/). The research of H.C.Pöhls was carried out in the project SEMIOTICS funded by EUs H2020 grant no. 780315.

References

1. Ateniese, G., Chou, D.H., de Medeiros, B., Tsudik, G.: Sanitizable Signatures. In: Proc. of European Symposium on Research in Computer Security (ESORICS 2005). LNCS, vol. 3679, pp. 159–177. Springer (2005)
2. Bilzhause, A., Pöhls, H.C., Samelin, K.: Position Paper: The Past, Present, and Future of Sanitizable and Redactable Signatures. In: Proc. of International Conference on Availability, Reliability and Security (ARES 2017). pp. 87:1–87:9. ACM (Sept 2017)
3. Biskup, J., Flegel, U.: Transaction-based pseudonyms in audit data for privacy respecting intrusion detection. In: International Workshop on Recent Advances in Intrusion Detection. pp. 28–48. Springer (2000)
4. Böhm, F., Menges, F., Pernul, G.: Graph-based visual analytics for cyber threat intelligence. Cybersecurity **1**(1), 16 (Dec 2018)
5. Brzuska, C., Pöhls, H.C., Samelin, K.: Non-Interactive Public Accountability for Sanitizable Signatures. In: Revised Selected Papers of European PKI Workshop: Research and Applications (EuroPKI 2012). LNCS, vol. 7868, pp. 178–193. Springer (2012)
6. Burkhart, M., Strasser, M., Many, D., Dimitropoulos, X.: Sepia: Privacy-preserving aggregation of multi-domain network events and statistics. Network **1**(101101) (2010)
7. Büschkes, R., Kesdogan, D.: Privacy enhanced intrusion detection. Multilateral security in communications, information security pp. 187–204 (1999)
8. Coppolino, L., D'Antonio, S., Mazzeo, G., Romano, L., Sgaglione, L.: How to Protect Public Administration from Cybersecurity Threats: The COMPACT Project. In: 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA). pp. 573–578 (May 2018). https://doi.org/10.1109/WAINA.2018.00147
9. Deutscher Bundestag: Gesetz zur Erhöhung der Sicherheit informationstechnischer Systeme (2015), https://www.bmi.bund.de/SharedDocs/Downloads/DE/Gesetzestexte/it-sicherheitsgesetz.pdf
10. Dewald, A.: Characteristic evidence, counter evidence and reconstruction problems in forensic computing. it - Information Technology **57**(6), 339–346 (2015)
11. European Commission: NIS Directive 2016/1148 (EU) of the European Parliament and of the Council (2016), http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016L1148

12. European Parliament and the Council of the European Union: Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC. Official Journal **OJ L 257 of 28.8.2014**, 73–114 (Jul 2014)
13. European Parliament and the Council of the European Union: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Official Journal **OJ L 119 of 4.5.2016**, 1–88 (May 2016)
14. Gartner Inc.: Security information and event management (siem) (2018), https://www.gartner.com/it-glossary/security-information-and-event-management-siem
15. Goldstein, M., Asanger, S., Reif, M., Hutchison, A.: Enhancing security event management systems with unsupervised anomaly detection. In: ICPRAM. pp. 530–538 (2013)
16. Höhne, F., Pöhls, H.C., Samelin, K.: Rechtsfolgen editierbarer Signaturen. Datenschutz und Datensicherheit - DuD **36**(7), 485–491 (Jun 2012), http://dx.doi.org/10.1007/s11623-012-0165-8
17. Jensen, M.: Challenges of privacy protection in big data analytics. In: 2013 IEEE International Congress on Big Data. pp. 235–238. IEEE (2013)
18. Johnson, R., Molnar, D., Song, D., Wagner, D.: Homomorphic signature schemes. In: Proc. of the RSA Security Conference - Cryptographers Track. pp. 244–262. Springer (Feb 2002)
19. Lanzi, A., Balzarotti, D., Kruegel, C., Christodorescu, M., Kirda, E.: Accessminer: using system-centric models for malware protection. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010. pp. 399–412. ACM (2010)
20. Latzo, T.: Efficient fingerprint matching for forensic event reconstruction (2020), under submission
21. Latzo, T., Freiling, F.: Characterizing the limitations of forensic event reconstruction based on log files. In: 2019 IEEE Trustcom/BigDataSE. IEEE (2019)
22. López, J., Oppliger, R., Pernul, G.: Why have public key infrastructures failed so far? Internet Research **15**(5), 544–556 (2005)
23. Menges, F., Böhm, F., Vielberth, M., Puchta, A., Taubmann, B., Rakotondravony, N., Latzo, T.: Introducing dingfest: An architecture for next generation siem systems. In: Langweg, H., Meier, M., Witt, B.C., Reinhardt, D. (eds.) SICHERHEIT 2018. pp. 257–260. Gesellschaft für Informatik e.V, Bonn (2018)
24. Miller, D., Harris, S., Harper, A., VanDyke, S., Blask, C.: Security information and event management (SIEM) implementation. Network pro library, McGraw-Hill, New York, NY (2011)
25. Miloslavskaya, N., Tolstoy, A.: New siem system for the internet of things. In: World Conference on Information Systems and Technologies. pp. 317–327. Springer (2019)
26. Mokalled, H., Catelli, R., Casola, V., Debertol, D., Meda, E., Zunino, R.: The applicability of a siem solution: Requirements and evaluation. In: 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE). pp. 132–137. IEEE (2019)
27. Nespoli, P., Gómez Mármol, F.: e-health wireless ids with siem integration. In: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC18), Barcelona, Spain. pp. 15–18 (2018)



28. Park, H.A., Lee, D.H., Lim, J., Cho, S.H.: Ppids: privacy preserving intrusion detection system. In: Pacific-Asia Workshop on Intelligence and Security Informatics. pp. 269–274. Springer (2007)
29. Parliament of the United Kingdom: Freedom of Information Act 2000. URL http://www.legislation.gov.uk/ukpga/2000/36/pdfs/ukpga_20000036_en.pdf (November 2000)
30. Pöhls, H.C.: Increasing the Legal Probative Value of Cryptographically Private Malleable Signatures. Ph.D. thesis, University of Passau (2018)
31. Pöhls, H.C., Höhne, F.: The Role of Data Integrity in EU Digital Signature Legislation - Achieving Statutory Trust for Sanitizable Signature Schemes. In: Meadows, C., Fernandez-Gago, C. (eds.) 7th International Workshop, STM 2011, Copenhagen, Denmark, June 27-28, 2011, Revised Selected Papers. Lecture Notes in Computer Science (LNCS), vol. 7170, pp. 175–192. Springer Berlin Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-29963-6_13
32. Pöhls, H.C., Höhne, F.: The Role of Data Integrity in EU Digital Signature Legislation - Achieving Statutory Trust for Sanitizable Signature Schemes. In: Revised Selected Papers from the 7th International Workshop on Security and Trust Management (STM 2011). LNCS, vol. 7170, pp. 175–192. Springer (2011), http://dx.doi.org/10.1007/978-3-642-29963-6_13
33. Pöhls, H.C., Samelin, K.: Accountable Redactable Signatures. In: Proc. of International Conference on Availability, Reliability and Security (ARES 2015). pp. 60 – 69. IEEE (Aug 2015)
34. Pöhls, H.C., Samelin, K., Posegga, J., de Meer, H.: Transparent Mergeable Redactable Signatures with Signer Commitment and Applications (MIP-1206). Tech. Rep. MIP-1206, Faculty of Computer Science and Mathematics (FIM), University of Passau (Aug 2012)
35. Putz, B., Menges, F., Pernul, G.: A secure and auditable logging infrastructure based on a permissioned blockchain. Computers and Security p. 101602 (2019)
36. Rieck, K., Holz, T., Willems, C., Düssel, P., Laskov, P.: Learning and classification of malware behavior. In: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. pp. 108–125. Springer (2008)
37. Schlette, D., Böhm, F., Caselli, M., Pernul, G.: Measuring and visualizing cyber threat intelligence quality. International Journal of Information Security (2020). https://doi.org/10.1007/s10207-020-00490-y, https://doi.org/10.1007/s10207-020-00490-y
38. Sgaglione, L., Mazzeo, G.: A GDPR-Compliant Approach to Real-Time Processing of Sensitive Data. In: Intelligent Interactive Multimedia Systems and Services. pp. 43–52. Springer International Publishing, Cham (2019)
39. Sobirey, M., Fischer-Hübner, S., Rannenberg, K.: Pseudonymous audit for privacy enhanced intrusion detection. In: Information Security in Research and Business, pp. 151–163. Springer (1997)
40. Stahlberg, P., Miklau, G., Levine, B.N.: Threats to privacy in the forensic analysis of database systems. In: Proceedings of the 2007 ACM SIGMOD international conference on Management of data. pp. 91–102. ACM (2007)
41. Steinfeld, R., Bull, L., Zheng, Y.: Content extraction signatures. In: Proc. of International Conference on Information Security and Cryptology (ICISC 2001). vol. 2288, pp. 163–205. Springer (2002)
42. The Apache Software Foundation: Apache http server project (2019), https://httpd.apache.org/
43. The National Archives: Redaction toolkit – editing exempt information from paper and electronic documents prior to release. URL http://www.nationalarchives.gov.uk/

documents/information-management/redaction_toolkit.pdf [last accessed: Nov. 2019] (Jul 2011)

44. United Kingdom Ministry of Justice: Lord Chancellor's Code of Practice on the management of records issued under section 46 of the Freedom of Information Act 2000. URL http://www.nationalarchives.gov.uk/documents/foi-section-46-code-of-practice.pdf [last accessed: Sep. 2019] (Jul 2009)
45. van Geelkerken, F.W.J., Pöhls, H.C., Fischer-Hübner, S.: The legal status of malleable- and functional signatures in light of Regulation (EU) No 910/2014. In: Proc. of the 3rd International Academic Conference of Young Scientists on Law & Psychology 2015 (LPS 2015). pp. 404–410. L'viv Polytechnic Publishing House (Nov 2015), https://drive.google.com/file/d/0B-Yu3Ni9z3PXM2lBajhCXzhoWk0/view
46. Vielberth, M., Menges, F., Pernul, G.: Human-as-a-security-sensor for harvesting threat intelligence. Cybersecurity **2**(23) (2019)
47. Vielberth, M., Pernul, G.: A security information and event management pattern. In: 12th Latin American Conference on Pattern Languages of Programs (SugarLoafPLoP 2018) (2018)
48. Wang, R.Y., Strong, D.M.: Beyond accuracy : What data quality means to data consumers. Journal of Management Information Systems **12**(4), 5–34 (1996), http://w3.cyu.edu.tw/ccwei/PAPER/ERP/dataquality%28JMIS%29.pdf
49. Williams, A.T., Nicolett, M.: Improve it security with vulnerability management. Technical Report - Gartner Inc. (2005)

6 DEALER: Decentralized Incentives for Threat Intelligence Reporting and Exchange

Current status:	Under Review	
Journal:	*Submitted to:* International Journal of Information Security	
Date of acceptance:	n/a	
Full citation:	Florian Menges , Benedikt Putz and Günther Pernul. DEALER: Decentralized Incentives for Threat Intelligence Reporting and Exchange. *Working Paper, University of Regensburg, 2020.*	
Authors contributions:	Florian Menges	45%
	Benedikt Putz	45%
	Günther Pernul	10%

Journal Description: The International Journal of Information Security is an English language periodical on research in information security which offers prompt publication of important technical work, whether theoretical, applicable, or related to implementation.

Noname manuscript No.
(will be inserted by the editor)

DEALER: Decentralized Incentives for Threat Intelligence Reporting and Exchange

Florian Menges · Benedikt Putz · Günther Pernul

Received: date / Accepted: date

Abstract The exchange of threat intelligence information can make a significant contribution to improving IT security in companies and has become increasingly important in recent years. However, such an exchange also entails costs and risks, preventing many companies from participating. In addition, since legal reporting requirements were introduced in various countries, certain requirements must be taken into account in the exchange process. However, existing exchange platforms neither offer incentives to participate in the exchange process, nor fulfill requirements resulting from reporting obligations. With this work, we present a decentralized platform for the exchange of threat intelligence information. The platform supports the fulfillment of legal reporting obligations for security incidents and provides additional incentives for information exchange between the parties involved. We evaluate the platform by implementing it based on the EOS blockchain and IPFS distributed hash table. The prototype and cost measurements demonstrate the feasibility and cost-efficiency of our concept.

Keywords Threat Intelligence Sharing · Blockchain · Smart Contract

1 Introduction

The threat landscape for IT infrastructures has grown steadily in recent years and this trend is continuing. At the same time, it is becoming apparent that the countermeasures currently available can hardly keep pace with the ongoing attacks. It has been shown that the exchange of threat information is an effective instrument for improving existing countermeasures and the overall situation. It leads to more knowledge about threats, earlier detection of attacks and thus to more effective countermeasures. The potential benefits of the threat information exchange have recently been recognized in the public sector by introducing corresponding legal regulations. For example, several countries already require the reporting of security incidents, especially for critical infrastructure operators.

While the exchange of threat information offers the aforementioned benefits for the security situation, it can also entail various disadvantages and problems that may prevent companies from participating. These include high additional costs for appropriately trained security personnel and infrastructure, possible data protection problems and the risk of publishing sensitive data. In addition to these problems, a complex set of legal reporting requirements must be taken into account. Companies must be able to provide non-repudiable proof of accurate reporting, both to avoid penalties and to potentially use the data as evidence in court. Consequently, sustained availability and integrity of the reported data must be ensured. Sharing platforms must address these problems by incorporating legal requirements as part of the design. Additionally, incentive structures must be created for the exchange of threat information, to offset costs and to motivate stakeholders to participate in the long term.

In doing so, we consider two use-cases separately. The platform intends to **1)** support the fulfillment of legally **obligatory reporting** and **2)** to create economic incentives for **voluntary reporting**. While these scenarios have different requirements and thus follow separate processes, the proposed platform option-

Florian Menges, Benedikt Putz, and Günther Pernul
University of Regensburg
Universitätsstr. 31, DE-93053 Regensburg
E-mail: firstname.lastname@ur.de

ally also enables sharing of obligatory reports. Based on these considerations, we formulate the research questions we intend to answer:

- **RQ1**: How can threat intelligence information be exchanged while ensuring availability, integrity and non-repudiation?
- **RQ2**: How can the exchange of threat intelligence information be incentivized?

To solve these problems, we propose a sharing concept and application prototype for a threat intelligence sharing platform based on Distributed Ledger Technology (DLT). DLT excels at providing *availability*, *integrity* and *non-repudiation* - the three requirements of RQ1. *Availability* is ensured by the underlying blockchain network, which consists of a large number of geo-distributed nodes maintaining a replicated ledger around the clock. At the same time, *integrity* assurance is provided through a sequentially linked hash chain, which ensures that the current world state is always the result of all past transactions. The consensus protocol assures that state transitions are append-only and previous entries are *non-repudiable.* Distributed Ledgers also provide verifiable decentralized execution of applications in the form of smart contracts, which also provide the option to implement digital currency in the form of blockchain tokens. These tokens can be used to provide decentralized *incentives* by assigning real value to threat intelligence information.

Existing work has attempted to address some of the aforementioned problems using DLT, however, the research questions have not been sufficiently addressed so far (see Section 2). For this reason, we propose the blockchain-based DEALER platform (**D**ecentralized Inc**E**ntives for Thre**A**t Inte**L**lig**E**nce **R**eporting and Exchange). It fulfills legal requirements for obligatory Cyber Threat Intelligence (CTI) reporting, while also providing an incentive structure to counteract possible participation drawbacks and to encourage voluntary sharing of CTI. Our contribution includes a novel protocol based on verifiers and token-based incentives to encourage fair sharing of high-quality threat intelligence data. To avoid trusting a third-party platform provider, the architecture is fully decentralized and maintained by independent blockchain operators and the participants themselves. In brief, the platform provides the following key features:

- **availability**, **integrity** and **non-repudiation** for obligatory reporting, fulfilling legal requirements
- decentralized **incentives** by leveraging blockchain tokens for purchase and sale of threat intelligence
- transactional **fairness** for both seller and buyer
- **quality assurance** through a verifier system

The remainder of this paper is structured as follows. In Section 2, we first provide an overview of approaches for platforms to report threat information, in particular with a focus on meeting the aforementioned security goals. In Section 3 we define requirements for the development of our plattform. Section 4 introduces our concept for the storage and incentivized exchange of threat intelligence information. In Section 5 we propose the system design for the application of our concept and present the implementation of our prototype. The cost structure and thus the practical feasibility of our prototypical implementation is evaluated in Section 6. The results of this paper are discussed in Section 7 and the paper is concluded in Section 8.

2 Related Work

The exchange of threat information has been the subject of practical and legislative work in recent years. These include laws in different legislations, such as the NIS Directive [1] in Europe and the IT-Sicherheitsgesetz (BSIG) [2] in Germany, which stipulate the reporting of incidents for providers of critical infrastructures. These legislations are also influenced by data protection requirements, which are for example specified by the General Data Protection Regulation (GDPR) [3] or the California Consumer Privacy Act (CCPA) [4]. At the same time, first platforms for the exchange of threat information are available. Examples are IBM X-Force [1] or Facebook threat exchange [2] as commercial platforms as well as MISP [3] and OPENCTI [4] as open source platforms. These platforms allow the exchange of threat information, however, data integrity or availability are not conclusively assured and incentive structures are not available. Central providers can advertise data integrity and availability, but ultimately it is always necessary to rely on the provider to ensure the protection goals are met. This is particularly problematic in the area of possible obligations to provide evidence, as manipulation of the data stock cannot be ruled out with central providers. At the same time, a single provider usually also represents a single point of failure when it comes to the availability of the platform. Furthermore, existing providers do not yet offer functionalities for quality-assured trading and thus an incentivised exchange of CTI information. A great deal of research has

[1] `https://eur-lex.europa.eu/eli/dir/2016/1148/oj`

[2] `https://www.gesetze-im-internet.de/bsig_2009/BJNR282110009.html`

[3] `https://eur-lex.europa.eu/eli/reg/2016/679/oj`

[4] `https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375`

been done on the requirements and challenges of implementing CTI platforms. In an early work, Serrano et al. [5] point out the fundamental problems for the exchange of threat information. Dandurand et al. [6] defined requirements for the exchange of information, emphasizing the necessity of assuring data integrity and availability, which is also supported by the work of Brown et al. [7]. Mohaisen et al. pointed out various open research questions in that field, such as possible dangers and negative incentives that may relate to the exchange of CTI [8]. In addition to this, there are also works that deal with specific implementations of CTI platforms, such as the MISP platform by Wagner et al. [3]. However, neither specific integrity or availability requirements nor the integration of incentives are considered. Literature also provides works that address the necessity of creating incentives for the exchange of CTI. Sauerwein et al. conducted an exploratory study that showed a need for incentivizing stakeholders within the exchange process [9]. This work is supported by Sillaber et al. examining the needs of stakeholders and resulting challenges [10]. While these studies provide possible starting points for the use of incentive procedures, the actual use of such procedures within CTI platforms is not considered. Moreover, there are also first approaches that try to implement CTI exchange on decentralized platforms. Alexopoulus et al. present a method for sharing security data streams based on a smart contract and data stream subscriptions [11]. Since the proposed data streams require a direct connection between the parties, the assurance of integrity and availability cannot be guaranteed. Incentive structures are also included in the work, but the design suffers from various weaknesses. Since the described on- and off-chain interactions of buyer and seller are independent of each other, negative consequences for fraud attempts during data transfer can only be implemented to a limited extent. In addition, the quality of the incident can vary during a stream, but only the entire stream can be evaluated by a buyer. This increases search costs on the marketplace because information about alerts is only available in aggregated form.

In summary, it can be stated that different works exist in the area of threat intelligence exchange that consider the requirements and the application of platforms. However, to the best of our knowledge, there is currently no work that allows an incentive-based, fair exchange of CTI information, while maintaining data integrity and availability to comply with regulatory requirements.

3 Objective and requirements

The exchange of threat intelligence information can be categorized into two different areas. On the one hand, unidirectional reports of security incidents, are stipulated by law and mostly concern companies that are relevant for the functioning of society. On the other hand, bidirectional exchange of security information between companies is done on a voluntary basis. The goal is an improvement of the information basis on security incidents for all participants and to increase their security level. The platform developed in this work aims to cover both use cases by enabling both reporting and exchange of security incidents. We consider the use cases **obligatory reporting** and **incentive-based exchange** of CTI information separately, as they should be independent features on the platform. However, a combination of both approaches should optionally be possible. There are different and unique requirements that result from each of these use cases, which are described in more detail below.

3.1 Requirements for reporting security incidents

The most important requirement for reporting security incidents is to comply with the underlying legal framework. For this reason, we first examine the implications of reporting obligations in more detail and derive resulting requirements for reporting. One basic requirement for a functioning reporting infrastructure is providing a way for a company to provide incident data and for legal authorities to receive this data. In this context, reports of security incidents are often time-critical, as legal authorities may have to react to reported events in a timely manner. For this reason providing very high **availability** is one key factor for the operation of a reporting infrastructure.

In the context of reports it is also of utmost importance to be able to prove who submitted a report. On the one hand, this is necessary so that authorities can take the necessary steps to prevent supply bottlenecks, for example. On the other hand, this also provides a guarantee for the reporting institution, as it enables it to prove that the reporting obligation has been fulfilled and thus avoid penalties. This necessity results in the requirement of **non-repudiation** and unambiguous assignment of reports. In addition, a further requirement results from the actual use of the data. Besides being used to prevent damage, the threat intelligence information obtained may also be used as evidence. Specifically, recorded data may either be used as evidence in court proceedings or as proof of damage against contractual partners such as insurance companies. Following this,

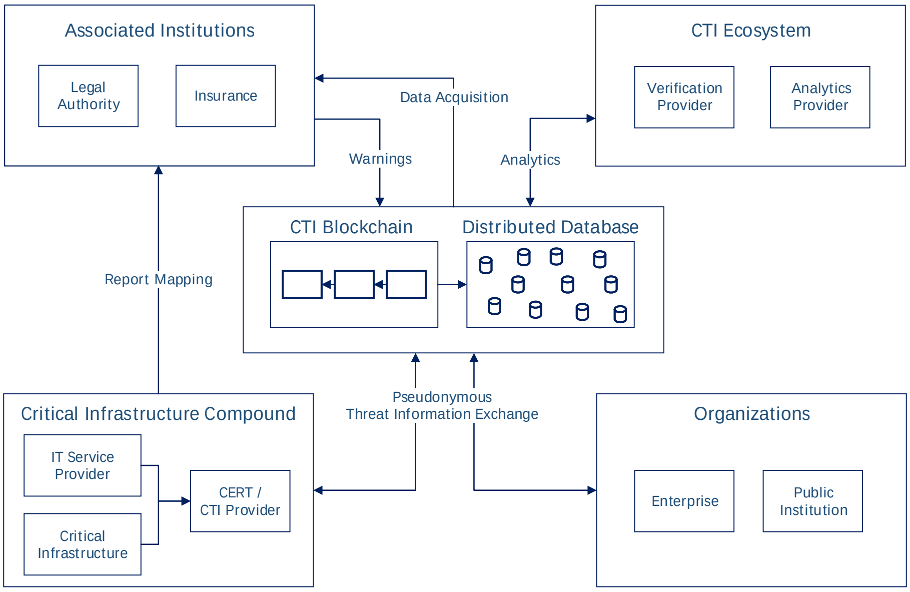


Fig. 1 High level overview of the DEALER threat intelligence sharing concept.

ensuring data **integrity** is an additional requirement in the reporting process that needs to be taken into account. Besides regulations that stipulate reports of security incidents, there are also regulations regarding the handling of personal data in different jurisdictions, such as the GDPR in the European Union. According to this, the platform must also provide the necessary tools to allow the protection exchanged data in compliance with legal regulations.

3.2 Requirements for an incentive system

In addition to requirements resulting from legislation, there are also functional requirements for exchange platforms. Every exchange of information on security incidents is accompanied by various risks. When publishing information, companies risk to accidentally leak important data. This may for example include company secrets or information about the company infrastructure, that may for example simplify attacks on that company. In addition, a reporting process involves costs for the collection, processing and dissemination of incident data. At the same time, the benefits of participating in an exchange platform are often difficult to quantify, especially with comparatively low legal penalties for omitted reports. From these points it can be concluded that companies tend to have little intrinsic motivation to report incidents themselves, whereas the motivation to passively obtain information from a reporting platform is likely to be high. As a result, an **incentive system** that motivates every participant on such a platform to actively participate can be defined as a further essential requirement for the sustainable functioning of such a platform (**RQ2**).

4 The DEALER sharing concept

In this Section we present the DEALER concept, which is designed to fulfil the previously defined requirements and to provide an incentives structure for sharing CTI information. This includes an ecosystem describing the stakeholders in the system, their roles and relationships and a marketplace describing the processes and concepts within the ecosystem, designed to guarantee sustainable CTI exchange. This Section provides an overview of the relationships within the system and the overall idea of the concept. The individual processes within the system are described subsequently.

The entire system, which is outlined in Figure 1 consists of five essential components. At the center of the system is a **blockchain** and a **distributed database**. These form the technological basis for the implementation of smart contracts, integrity-secured storage of exchange processes and provide decentralized storage structures for reported security incidents. The starting point for reports within this system are **Critical Infrastructure Compounds**. These include the critical infrastructure operator, an IT service provider if applicable, and a CTI provider. The CTI provider takes care of external communication and acts as a so-called contact point, a construct that can be derived from legal requirements for incident reporting. The information collected is intended for either Associated Institutions or Organizations. **Associated Institutions** describe participants who are interested in the reported information within the scope of reporting obligations. These can for example be legal authorities to which a reporting obligation exists. These can also be other institutions, such as insurance companies, to which a possible claim can be made accessible via the platform. On the other hand, there are **Organizations** that are not affected by reporting obligations, but are nevertheless interested in participating, for example to increase their own level of protection. Analyses and services within the system are provided by the **CTI ecosystem**. This enables external service providers to bring their services into the system. For example, verification providers can offer qualitative incident data evaluation, or analytics providers can aggregate information on several incidents and offer it within the system.

DEALER's overall concept defines two central use cases: statuatory incident reporting and incentive-based threat intelligence exchange. Both concepts are briefly described below before we take a closer look at the underlying processes.

Obligatory reports are generated by the Critical Infrastructure Compound and transferred to the blockchain. The transmitted data is pseudonymized and encrypted in such a way that only the receiving authority can access it. In connection with such a report, the data can also be made available to other users of the platform as part of the incentive-based exchange. However, this step is explicitly optional and must be actively selected.

The **incentive-based exchange** process is based on an economic model, where participants can offer and demand information on security incidents. For this purpose, a separate token is introduced on the platform, which functions as an internal currency and is used as economic reward for active participants. When threat information is provisioned, structured incident data is transferred to the blockchain in encrypted and pseudonymized form. The information provided can then be sold to other participants or made available as a report. The uploaded incident information is assigned to verifiers who ensure its data quality against a fee. After successful verifications, the data can be traded on the platform at the previously defined price.

In addition to these two sharing mechanisms, legal authorities may additionally issue global warnings regarding threats to all participants. In some legislations, such as the IT security law in Germany [12], such global warnings are part of the reporting obligation and thus necessary for compliance. The warnings also represent an additional benefit for the platform participants: the free CTI provided by the legal authority supplements purchasable incident information.

After this high-level introduction to the basic concept of DEALER, the core processes of the platform are presented in more detail below. They include Registration (4.1), Sharing (4.2), Verification (4.3), Purchase (4.4) and Fairness (4.5).

4.1 Registration

Initially, participants must register to be able to transact on the decentralized marketplace. Each participant has an account with a balance of fungible tokens, which may be used to trade incidents. To prevent sybil attacks, we require a fixed initial token stake s_i to create the participant's balance. This prepayment requires a meaningful investment, while not deterring new users. The user balance is managed by the platform. Withdrawals are allowed on request up to the initial fee, which must remain until the participant closes the account.

Verifiers are treated separately during registration, as they are given free access to incident information and must evaluate it. The purpose of registration is to achieve a unique identification of the verifier, for example by requesting a tax number, identity documents or a social security number. This registration process is intended to prevent the risk of verifier misuse (i.e. free-riding or submitting default ratings). In contrast to regular participants of the platform, verifiers must be approved before participating in the verification process. During bootstrapping of the verifier pool, approval can be conducted by the platform developer. Once the verifier pool has reached the minimum size (see Section 7.2), new participants can be approved through majority votes of existing participants.

Additionally, the platform provides an exclusion option for malicious verifiers. Exclusion of a verifier must be approved by a majority of the verifier pool through multisignature votes. Any verifier may initiate such a vote by providing evidence for several instances of misbehavior (i.e. repeatedly submitting default or unrealistic verification reports).

Besides preventing misuse, the goal of authenticating verifiers is to ensure an intrinsic interest in the analysis of security incidents and possession of the necessary technical expertise for actual incident information assessments. Appropriate verifiers could for example be

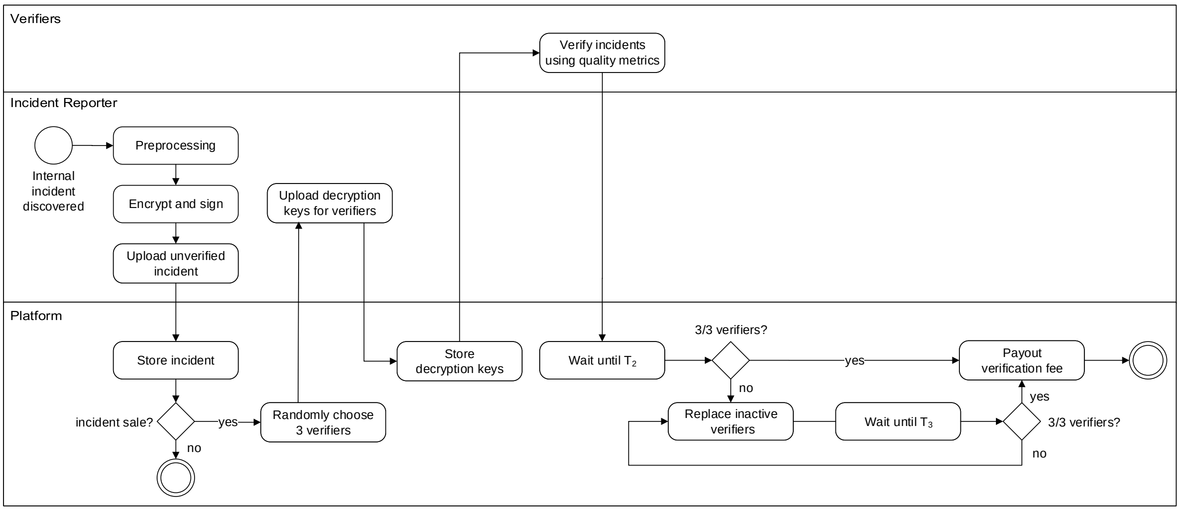


Fig. 2 Incident sharing process on the DEALER platform.

threat intelligence vendors, CERTs or security operations professionals.

4.2 Sharing

Figure 2 shows a BPMN diagram of the sharing process from incident detection to data upload, verification and provisioning on the platform. Initially, the participant locally performs required preprocessing steps. These include anonymization (removing private data and identifying details), addition of public descriptive metadata and encryption of the incident with a symmetric key k. The metadata also includes a sale price p_s. A signed transaction is submitted to the platform and the incident is uploaded to the distributed database. If the participant decides to sell the incident to other users, a verification fee p_v must be paid once with the initial transaction (i.e. $p_v \sim 0.6p_s$). The incident is then made available on the marketplace and verification is initiated. Three random verifiers are chosen from the verifier pool. The seller then uploads three keys $k_{v1}/k_{v2}/k_{v3}$ for each chosen verifier, encrypted with each verifier's public key, and notifies the platform at time T_1. The verifiers retrieve and decrypt the uploaded incident with their individual key file. They assign an initial rating value based on a set of platform-provided quality metrics (see Section 4.3).

The verifiers submit the verification result to the platform. If all results arrive until time T_2, the verification fee p_v is distributed equally among the verifiers (i.e. $\frac{p_v}{3} \sim 0.2p_s$ per verifier). If any verifier does not respond, the seller may trigger a replacement of inactive verifiers. These verifiers must respond until time T_3 ($T_3 > T_2 > T_1$), else the seller may request a removal of the incident from the platform and partial reimbursement of the verification fee ($\frac{p_v}{3}$ per missing verifier).

For obligated incident reporting, the participant may want to keep the incident confidential and not share it with verifiers. In this case, the participant only uploads a key for the regulatory authority and no verification is performed. The platform provides a timestamp and proof of reporting for the incident.

4.3 Verification

The data quality verification conducted by verifiers serves as an incident reputation bootstrapping mechanism. We propose a 5-point rating scale for incident quality from 1 (very low) to 5 (great). The verification needs to be as objective and meaningful as possible to provide guidance for buyers, since the actual data is encrypted. The following items serve as verification guidelines:

- consistency with metadata of the seller's previous incidents
- similarity check for incident metadata and verified incidents
- assessment of various threat intelligence quality indicators [13]

After receiving the incident data, each verifier independently performs a verification of the contained information. A basic consistency check using metadata of the seller's previous incidents verifies that the incident

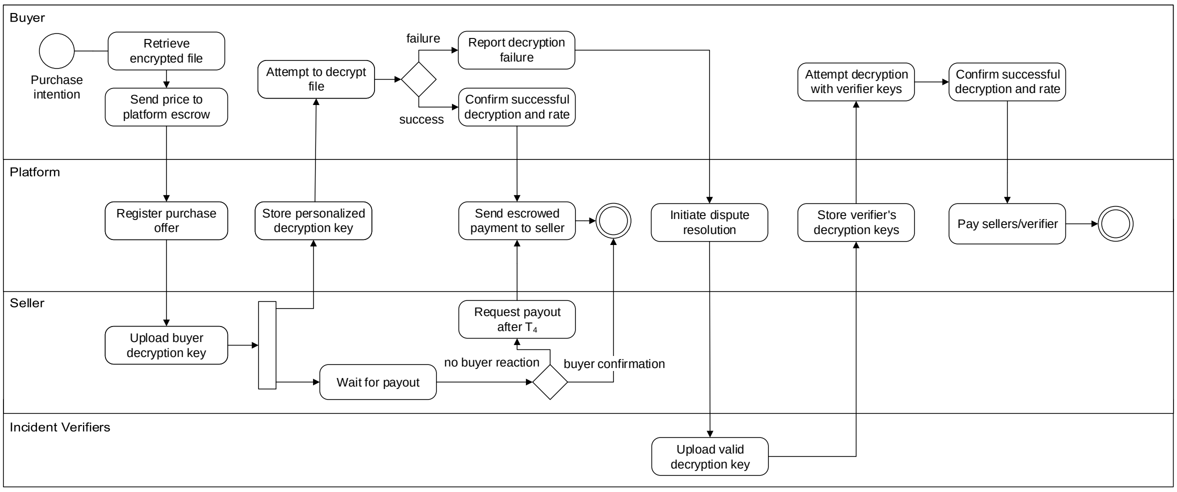

Buyer
Purchase intention
Retrieve encrypted file
Send price to platform escrow
Attempt to decrypt file
failure
success
Report decryption failure
Confirm successful decryption and rate
Attempt decryption with verifier keys
Confirm successful decryption and rate
Platform
Register purchase offer
Store personalized decryption key
Send escrowed payment to seller
Initiate dispute resolution
Store verifier's decryption keys
Pay sellers/verifier
Seller
Upload buyer decryption key
Request payout after T$_4$
no buyer reaction
buyer confirmation
Wait for payout
Incident Verifiers
Upload valid decryption key

Fig. 3 Purchasing process on the DEALER platform.

originates from the same industry. To avoid duplicates and resold incidents (see also Section 7), verifiers compute a similarity score to other previously downloaded incidents (i.e. using the *simhash* algorithm [14]). For apparent duplicates, verifiers then submit a low score without additional quality assessment.

Regarding threat intelligence quality indicators, the platform provides a structured assessment process. This procedure is intended to help verifiers make objective and comparable assessments of security incidents by iteratively processing predefined questions.

To achieve this, the implemented questions are based on objective CTI data quality indicators developed for STIX2 [13]. The quality criteria are divided into three major domains. These include information about the contained data, object representations within the data and the completeness of the available information. In particular, the data model domain reflects information about the *representational consistency* of the data representations and the *concise representation* of the stored information. The object metrics area considers the *objectivity* of the data collected as well as metrics about the *relevancy* of the stored data regarding the situation described. The third domain addresses the completeness of the available information in more detail. This includes the examination whether an *appropriate amount of data* is used to convey the facts presented. In addition to this, the *syntactic accuracy* of the data transported as well as the *schema completeness* of the data is checked.

4.4 Purchase

The incident purchase in Figure 3 process starts off with a potential buyer browsing the repository of previously uploaded incidents. For this purpose the platform frontend offers sophisticated search and filter functionality. Metadata and ratings are provided for each individual incident by verifiers and past purchases. Once an incident of interest has been identified for purchase, the buyer retrieves the encrypted incident to verify its availability. If the incident is available, the buyer places an order for the incident and pays tokens corresponding to the sale price p_s to the platform escrow. The seller accepts the order by uploading a public key encrypted file key k_b for decryption. In case of successful decryption, the buyer notifies the platform by sending a confirmation along with an incident rating.

If the decryption fails, the buyer notifies the platform about the failure, which initiates the dispute resolution process. Any verifier may then provide an independent copy of the decryption key to the buyer. In the unlikely event that the buyer is still unable to decrypt the file, keys must be uploaded by additional verifiers to resolve the dispute. Once the buyer is able to decrypt the file, the buyer submits a rating for the incident and closes the dispute. Each contributing verifier receives an equal share of dispute fee p_d as compensation, which is deducted from the sale price (i.e. $p_d \sim 0.06p_s$). The dispute fee should be proportionately low for two reasons: verifiers can upload key copies in an automated fashion; and sellers should not lose a disproportionate amount of the sale price in case of unwarranted disputes.

A time lock T_4 is in place to allow parties to redeem their tokens if the counterparty fails to respond. If the buyer does not report decryption success or dispute, the seller may collect the sale price after T_4 has expired. If the seller never accepts the offer, the buyer may redeem the locked tokens after T_4.

4.5 Fairness

Fairness of incident purchase must be considered from two perspectives:

- **Seller Fairness**: An honest seller is guaranteed to receive the advertised sale price for providing a correct decryption key.
- **Buyer Fairness**: An honest buyer is guaranteed to receive the cleartext of the purchased incident, or is refunded the deposited purchase price.

We guarantee Fairness based on the following assumption: There is always at least one honest verifier that provides a valid decryption key. After verification, there are at least four copies of the decryption key (the seller and three verifiers) available on the platform. It is reasonable to assume that there is at least one honest participant among these four, which provides a decryption key in case of an issue with the seller's key.

We now analyze the various ways how seller and buyer may attempt to cheat, and how the protocol mitigates these attempts.

Buyer Fairness. The honest verifier assumption means that the buyer will always receive a decryption key, and that there is no scenario where the buyer will not be able to decrypt the file. Conversely, the buyer will also not receive the deposited price back. In case the seller attempts to cheat by uploading a wrong decryption key for the buyer, the buyer can initiate a dispute to receive a correct key from a verifier. Verifiers receive a dispute fee p_d as participation reward for uploading correct keys during a dispute. The seller is thus disincentivized to send wrong keys, since that increases the likelihood of a dispute and results in a loss of p_d tokens.

In case both seller and verifier keys are incorrect, the buyer may be unable to decrypt the item at all. This will not occur in practice based on the assumption that the majority of verifiers is honest and provides correct keys. This assumption can be made based on two properties of our platform:

1. random assignment of verifiers to incidents makes seller-verifier pairings unlikely, and repeated collusive arrangements are time-consuming
2. misbehavior is disincentivized through significant verifier registration requirements (see Section 4.1) coupled with the possibility of exclusion

We have thus ensured that the seller is punished for uploading wrong key material, while the buyer is able to decrypt the purchased file. To increase the buyer's confidence in receiving a correct key, the time of last platform activity of an incident's verifiers can be shown in the user interface.

Seller Fairness. The buyer may attempt to cheat the seller by not responding after the seller has provided the decryption key. For this reason, there is a deadline for the buyer to respond, which starts from the time the seller has uploaded the key and ends after time T_4. If there is no response after expiry, the seller may redeem the purchase price.

The buyer may also collude with the verifiers to falsely vote for seller misbehavior. In this case the honest seller would lose out on p_d tokens deducted from the sale price. This scenario is unlikely, since the buyer has no incentive to collude with verifier. If buyer and verifier are in contact, they could exchange data and tokens through another channel with a reduced price. In practice, this is unlikely to occur, since there is a large overhead for buyers to contact verifiers for every incident they are interested in.

If not colluding with a verifier, the buyer has no incentive to blame the seller. He cannot receive any tokens back that were paid for the sale, and he is guaranteed to receive a correct decryption key if at least one verifier is active.

These considerations guarantee Seller Fairness, with the restriction that the seller may lose out on a small portion of the sale price p_d in case of a dispute. Disputes cannot be prevented by the seller, but buyers have no incentive to start disputes, so we expect them to be negligible in practice.

5 Application Prototype

To implement the sharing concept, we choose a combination of blockchain technology and distributed hash tables. This avoids having to trust a single third-party service provider to provide storage and confidentiality. A data storage distributed in this way can be maintained collaboratively and only by participants interested in sharing data. Blockchain networks also allow utilizing virtual currencies that provide possibilities to realize built-in sharing incentives for participants. In the following we first discuss the technologies used for our prototype (5.1). Subsequently, we develop the conceptual architecture (5.2) and briefly present our prototypical implementation of the sharing platform (5.3).

5.1 Technology selection

In this section we will first discuss the underlying technologies for our sharing platform. This includes the permission model, the approach for storing incident data as well as the chosen blockchain platform.

Permission Model: Blockchain frameworks can be categorized as permissioned or permissionless, depending on whether the validators must be authenticated or not. Thus, we first choose a suitable permissioning model for our sharing concept. Permissioned blockchains are attractive due to increased control of the platform by its operators, the independence from public blockchains and zero transaction fees. However, the operation of a Permissioned Blockchain requires that the infrastructure is operated by the participants themselves. This results in high initial costs, while availability is only guaranteed by a limited number of blockchain nodes. It is also still unclear how digital tokens can be created and exchanged for fiat currency in a permissioned setting. In a permissionless setting, the blockchain nodes and infrastructure are already available, but fees must be paid to the maintainers of the platform. Public blockchains usually provide a high number of distributed nodes that guarantee high availability, while token distribution can be handled transparently using existing exchanges. Since high availability and incentives for participants are essential aspects of our concept, we choose a *permissionless* blockchain approach for our concept.

Blockchain Platform: Commonly, researchers use Ethereum for permissionless blockchain application prototypes due to its good tool support and large developer community [15]. Although the design can also be implemented with other permissionless blockchain systems, we choose the EOS blockchain[5]. First and foremost, EOS does not charge users transaction costs. Transaction allowances are determined based on staked EOS tokens, thus lowering the long-term cost of using the platform. In addition, EOS provides more scalability regarding transaction throughput (up to 8,000 transactions/second compared 15 transactions/second for Ethereum [16,17]).

Data Storage: Due to high costs associated with smart contract data storage, larger data items are commonly stored off-chain in blockchain applications [18]. One way of trading data using blockchain is settling the trade on-chain and trading the actual data off-chain [11]. This avoids the need for another storage platform besides the blockchain. However, it also requires the seller to re-upload data to every buyer, which means that both seller and buyer need to be online at the same time. A decentralized off-chain storage platform avoids this issue. To ensure an integrity link between the blockchain network and the off-chain store, the database should be content-addressable. Since only encrypted information is stored off-chain, access control is not required. Distributed Hash Tables (DHTs) provide these properties: they offer public, distributed and content-addressable key-value data storage. We opted for IPFS[6] as the DHT implementation in the prototype. IPFS is widely used in research as an off-chain storage solution, and it provides the features needed for sharing CTI data and encryption keys.

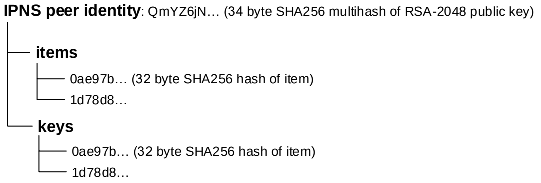

IPNS peer identity: QmYZ6jN... (34 byte SHA256 multihash of RSA-2048 public key)
items
0ae97b... (32 byte SHA256 hash of item)
1d78d8...
keys
0ae97b... (32 byte SHA256 hash of item)
1d78d8...

Fig. 4 IPFS off-chain storage folder hierarchy (for each user).

One of these features is a fixed address for each peer for sharing dynamic content (referred to as *IPNS address*). In the DEALER prototype, each seller and buyer operates a IPFS node. The node's IPNS address is based on the hash of the peer's public key and can only be updated with a signed update from that peer. We exploit this functionality to statically address each user's shared incidents and decryption keys. We leverage the IPFS Mutable File System to create a local folder hierarchy corresponding to the files we intend to share (see Figure 4). The root hash of this folder hierarchy changes every time an item or key is added to a folder. Each time that happens, the updated hash is published to the peer's IPNS address. Other peers can resolve this address to retrieve the latest incidents and keys shared by other users. By *pinning* content hashes, verifiers permanently replicate the encrypted incident shared by the seller to ensure its availability. Verifiers are incentivized to replicate seller content, since they potentially profit from each sale in case of a dispute (see Section 4.5).

5.2 Architecture and data model

As shown in Figure 1, the prototype architecture consists of a smart contract on the EOS blockchain platform and IPFS based decentralized storage. The blockchain platform provides executable smart contracts that implement the *Platform* role in the processes

[5] https://eos.io

[6] https://ipfs.io

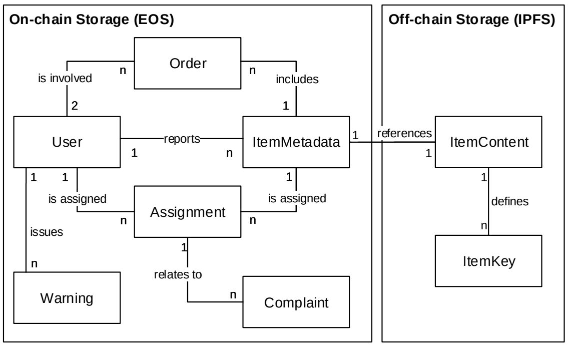


Fig. 5 Simplified Entity Relationship Model of data stored on the DEALER platform.

described in Section 4. IPFS provides storage capabilities for reported incident data and encryption keys. It also provides pseudonymous identity: Participants sign up with blockchain accounts, which are authorized through public-private key pairs and represented by unique addresses. Figure 5 gives an overview of the platform's data model.

The model shows a distinction between *on-chain* and *off-chain* storage. The on-chain storage manages transaction information and metadata including assignments of users (*User*), reports (*ItemMetadata*), votes *Assignment* and purchases (*Order*) of incident data. The off-chain storage holds the actual incident data (*ItemContent*) as well as the encrypted decryption keys for the information (*ItemKey*). The *ItemMetadata* table contains the reported incidents' metadata, including a short description, the originating industry, the price and a reference to the reporting user. *ItemMetadata* also contains the CTI item's hash, which links the metadata to the full incident data *ItemContent* stored off-chain. Using the hash reference, data can be retrieved from IPFS through a DHT lookup and verification of the retrieved file against its hash reference. The assignment of randomly selected verifiers is done using the *Assignment* table by establishing a link between the verifying user and the respective item. This table also stores the results of item verification, while cumulative results of verification and rating processes are stored in the *ItemMetadata* table. The assignment table is additionally linked to the *Complaint* table, which stores complaints about inaccurate verifications. The *Order* table finally contains the transactions associated to an order, where a transaction establishes the relationship between the buyer and the seller, as well as the item concerned. Besides storing report items, the application also allows the issuance of warnings. These can be inserted by authorities as a specific type of user and stored in the table *Warning* on chain.

5.3 Application Prototype

The prototypical implementation of the platform consists of three major components: the smart contract on the EOS blockchain based on EOS C++ code, the IPFS data storage and a DApp (Decentralized Application) front end based on Node.JS. Since Smart Contract and data storage were already described previously, this section focuses on the implementation of the DApp. Figure 6 shows the user-interface of the DApp, which is explained in more detail in the following.

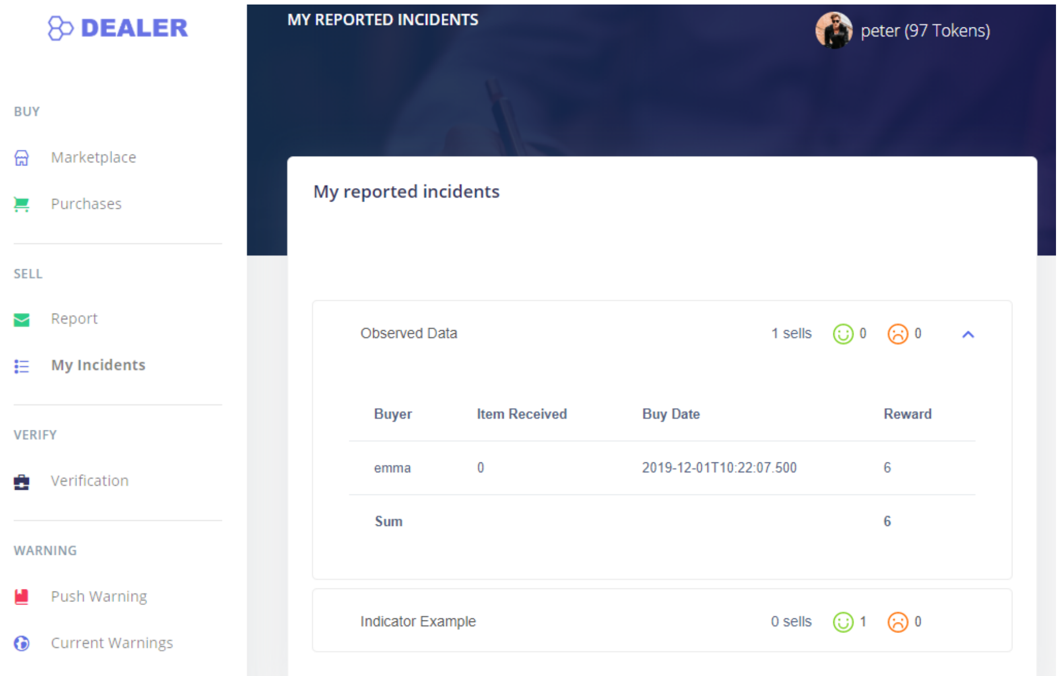
The application's user interface offers four fundamental areas tailored to each participant type. The area *BUY* allows potential buyers to get an overview of offers on the platform and to buy and download available incident information. The overview contains a short description of the incident information as well as its current verification status and price. Buyers can also manage past purchases and re-download previously bought information at any time. The area *SELL*, allows sellers to report an incident to the blockchain. Such a report can contain a title, a short description, the corresponding industry sector, the actual incident data and a sale price. Incidents are encrypted using AES-256-CBC before being uploaded to IPFS. After the DHT upload, the hash reference and metadata are submitted to the smart contract. If the incident was intended for sale, RSA-encrypted copies of the AES symmetric keys are shared with the verifiers using their public keys stored on the blockchain. Besides reporting, sellers can manage past reports and view the verification status and number of their successful sales.

The *VERIFY* section allows the user to act as a verifier for an incident. The verifier is presented with a list of all incidents assigned for verification. For each individual incident, the verifier is presented with a wizard as shown in Figure 7. The wizard sequentially requests input for the quality criteria defined in Section 4.3. The verification results are arithmetically averaged after submission and sent to the platform in a blockchain transaction. Although the prototypical application allows a weighting of the individual quality criteria, this was not implemented within the demonstration prototype for reasons of clarity.

Finally, the area *WARNING* allows authorities to issue warnings on current threats to platform participants. Warnings contain informational text and structured incident information for particularly dangerous threats.

The source code for the prototype can be downloaded at the project repository[7]. A live version of the

[7] `https://github.com/Dealer-Platform/`



DEALER

BUY

Marketplace

Purchases

SELL

Report

My Incidents

VERIFY

Verification

WARNING

Push Warning

Current Warnings

MY REPORTED INCIDENTS

peter (97 Tokens)

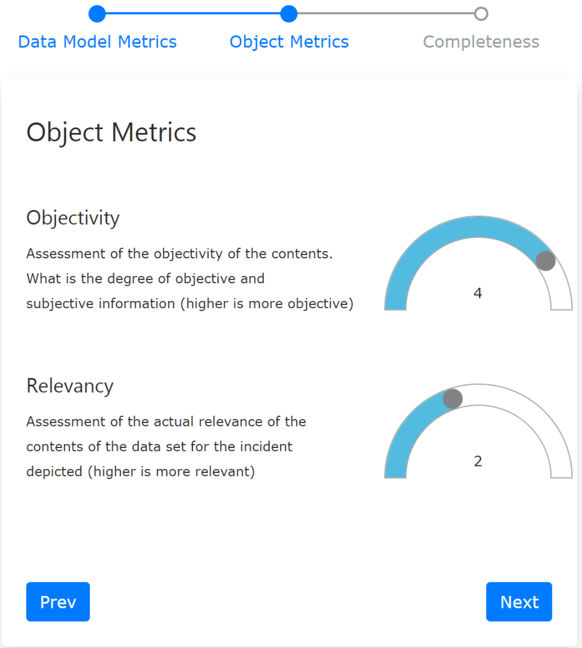
My reported incidents

Observed Data 1 sells ☺ 0 ☹ 0

Buyer	Item Received	Buy Date	Reward
emma	0	2019-12-01T10:22:07.500	6
Sum			6

Indicator Example 0 sells ☺ 1 ☹ 0

Fig. 6 Prototypical implementation of the DEALER platform.



Data Model Metrics — Object Metrics — Completeness

Object Metrics

Objectivity

Assessment of the objectivity of the contents. What is the degree of objective and subjective information (higher is more objective)

4

Relevancy

Assessment of the actual relevance of the contents of the data set for the incident depicted (higher is more relevant)

2

Prev

Next

Fig. 7 DEALER verification wizard.

DApp is available online[8], and the deployed EOS contract can be inspected on the EOS Kylin testnet[9].

6 Evaluation

After presenting the prototype design, we now evaluate whether the chosen blockchain platform fits the needs of threat intelligence reporting. Since EOS supports > 1000 transactions per second [17], we do not expect throughput to become a bottleneck. However, there are costs associated with transacting on a public blockchain, which should be evaluated in more detail.

Smart contracts on EOS require CPU, NET and RAM to execute. CPU and NET represent the processing and network utilization of transactions and are acquired by staking EOS for a fixed time. RAM is needed to store data in the smart contract state and is purchased at a fixed price. To calculate the required stake per user to run the contract sustainably, we evaluate the resources consumed by our smart contract in Table 1. Transactions were run multiple times with differing parameters on the EOS Kylin testnet. For CPU/NET, the values represent locked currency, i.e. to share one incident per day, EOS worth 0.2€ must be staked permanently. For RAM, the costs cumulate with each executed action and are thus much higher. For this reason we now focus on RAM costs.

In the following we estimate the costs of the platform based on a real-world example. Therefore, we assume that the platform will be used for the reporting obligations of critical infrastructures in Germany. Ac-

[8] `https://dingfest.ur.de/dealer/`
[9] `https://kylin.bloks.io/account/eosdealeradm`

Table 1 Resources consumed by the EOS smart contract. EOS price: 2.00€, RAM price: 0.058 EOS/kb, CPU cost: 0.05 EOS/ms, NET cost: 0.001 EOS/kb

Action	**CPU** (stake)		**NET** (stake)		**RAM** (purchase)	
Sharing	1.76 ms,	0.201€	0.256 kb,	0.0005€	0.755 kb,	0.088€
Verification	0.58 ms,	0.067€	0.120 kb,	0.0002€	0.000 kb,	0.000€
Purchase	1.07 ms,	0.065€	0.112 kb,	0.0002€	0.153 kb,	0.018€
Warning	0.74 ms,	0.084€	1.71 kb,	0.0034€	1.896 kb,	0.221€

cording to the Federal Office for Information Security (BSI), it is estimated that around 250 reports are carried out annually in 9 industry sectors [19]. The EOS RAM needed to store 250 incidents costs 22€ per year at the current conversion rate. The verifications do not cost any RAM since they only modify storage entries and don't add data.

We assume that participating companies are particularly interested in information from their sector (on average 28 reports per sector). According to the BSI, 1648 institutions in Germany are currently affected by the reporting obligation [19]. We thus estimate about 1,648 * 28 = 46,144 purchases to be made in ongoing operations (823€). Additionally, we assume that authorities may issue warnings about once a month (3€). In summary, we expect a total RAM cost of 848€ to store all platform interactions occurring in one year. This is quite a feasible amount, considering that it covers more than a thousand institutions.

7 Discussion

In this Section the results of this work are discussed. For this purpose, Section 7.1 reviews the previously defined requirements and compares them to the actual results achieved in the prototype. Subsequently, we discuss security concerns for the platform in Section 7.2.

7.1 Requirements

Reporting Requirements. At the beginning of this work, Section 3 defined various requirements for a platform that simultaneously complies with legal requirements and offers incentives for the exchange of CTI information. Specifically, the *integrity* and *availability* of data as well as the *non-repudiation* of reports were defined as target values for compliance with legal requirements. The decentralized blockchain technology used provides the necessary basic conditions to build a platform that is compliant with these requirements. One of the most important features of a blockchain is the assurance of data integrity using the decentralized ledger technology. Our solution assures *integrity* by including a hash of the data on-chain. Due to the EOS blockchain's immutability, this hash can be traced back to the original upload transaction and authenticated with the sender's signature.

Furthermore, blockchains also offer a very high *availability* of the network nodes as pointed out by Weber et al. [20]. The main restriction of blockchain systems pointed out in their work is the restricted write-access availability. However, this is mainly the result of the low number of possible transactions per second of the considered blockchains Ethereum and Bitcoin. Since the EOS blockchain exceeds the possible transactions per second of these networks by orders of magnitude [17], restrictions of write availability are unlikely. It should also be emphasized that the EOS network is distributed over the entire globe [10], which makes the availability of the network relatively independent of local events.

In the presented prototype, metadata of each reported security incident is stored on the EOS blockchain in a publicly accessible manner. In order to establish a reference for *non-repudiation*, a timestamp is included in the incident metadata, which proves the report's existence. A reference to the reporting EOS wallet is included to link the report to the reporter's EOS wallet. The full incident data is stored on the IPFS DHT and replicated by the incident's seller and verifiers, ensuring *availability* of off-chain data through sufficient redundancy.

In addition to this, the prototype also provides the necessary tools to protect personal data within reports according to legislations such as the GDPR. To achieve this, the exchanged information is processed in an encrypted form on the platform. Each data flow is addressed to an explicit recipient and protected with the corresponding public key. This ensures that only the receiving authority can view the reported information. In the case of an exchange on the marketplace, the data is also encrypted and assigned to a buyer and verifier as specific recipients. However, since the data is transferred to different recipients, the mere assignment to the recipient is not sufficient for information and pri-

[10] https://glass.cypherglass.com/map/main/top50

vacy protection. According to this, the offering company must decide here which data may be passed on to recipients. Both the interests of the company and the legal situation must be taken into account.

Incentives. At the same time, it was shown above that incentives are a necessary condition for an active exchange between the parties involved. In order to be able to implement such incentive procedures, a marketplace was created within the platform for the mutual exchange of CTI information. Participants can offer their incident information at the marketplace in return for payment. This gives them the ability to compensate costs incurred in the detection and recording process and thus provides a financial incentive to participate in the platform. Another focus of the platform is to ensure sustainability of the implemented incentive structure. Verifiers ensure the data quality of the traded CTI information as well as functions that guarantee transactional fairness for both buyer and seller. Verifiers and sellers are incentivized to host incident data on IPFS since they profit from incident sales.

Advantages over traditional CTI sharing platforms. Overall, it can be concluded that the platform for the exchange of CTI information presented in this work offers several specific advantages over existing CTI sharing platforms. Traditional systems usually rely on trust in a Trusted Third Party (TTP) to implement the data protection goals. In contrast to this, the decentralized DEALER system guarantees these protection goals without the need for a specific trust relationship. The availability of the platform is distributed among different independent actors and no central actor is required for integrity proofs. Moreover, the implemented marketplace for the exchange of information is likewise not dependent on the trustworthiness of actors. Within the implemented smart contract, the sales process as well as the selection of verifiers is predefined and transparent for all participants.

7.2 Security

Free-riding verifiers. An important consideration is prevention of free-riding verifiers. Every verifier periodically receives free access to a randomly selected incident. As a result, verifiers should be punished if they do not perform verification as requested. If a verifier repeatedly fails to verify assigned incidents in active status, other verifiers may start a multisignature vote for verifier removal. This should encourage verifiers to only remain active when they intend to verify, to avoid losing their verifier status.

Content reselling. Reselling information is a common concern for data marketplaces [11]. On our platform, the hash of the shared incident data is stored on the blockchain and thus the identity of the original author can be clearly established through the timestamp and the signing public key of the transaction. While uploading duplicate hashes is prevented by the smart contract, resellers can slightly modify the incident to change its hash. However, in the long run the similarity checks introduced in Section 4.3 should help recognize duplicates. If a duplicate is recognized, verifiers may submit a low rating. Similarly, buyers are likely to notice that they received a duplicate and rate the incident poorly, leading to a decreasing rating. This should discourage potential buyers and lead to decreasing profits from reselling attempts.

Sybil attacks. Sybil attacks involve attackers being able to create new identities cheaply to manipulate the application. They can be mitigated by introducing nontrivial barriers to entry. On the DEALER platform, this threat mainly applies to sellers and verifiers. Sybil sellers could flood the platform with incidents to overwhelm verifiers. Sybil verifiers could dilute the quality assurance verifiers are supposed to provide. Therefore, as established in Section 4.1, both sellers and verifiers need to deposit cryptocurrency to create an account. Verifiers additionally need to establish their organizational affiliation on registration. These measures should deter any attempt at Sybil attacks.

Incident confidentiality. A compromise of the RSA or AES encryption schemes may void the confidentiality of incidents stored on IPFS. In that case we assume there is sufficient advance notice for sellers to re-encrypt their incidents with a another unbroken encryption scheme. Sellers and verifiers are then incentivized to stop hosting compromised incidents on IPFS, since they no longer profit from sales if anybody can decrypt the incidents without purchase.

Verifier collusion. The platform requires a minimum number of verifiers to ensure their assignment is sufficiently random to deter collusion. If assignment is not random, sellers may collude with verifiers to ensure incident verification. Alternatively, a pair of verifiers

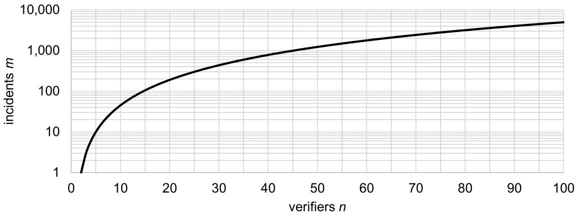


Fig. 8 A pair of verifiers is assigned to the same incident every m incidents, given n verifiers.

may collude during dispute resolution. The binomial coefficient determines the probability of assigning two verifiers to the same incident (n is the number of verifiers, and $k = 2$):

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

As shown in Figure 8, with 15 verifiers the probability is $< 1\%$, while with 50 verifiers it is $< 0.1\%$. Hereby we determine 15 verifiers as a safe minimum number of verifiers to safely operate the platform. Since verifiers may be temporarily inactive, a higher number is preferable in practice. With an expected amount of 250 reports annually (see Section 6), each pair of verifiers shares only 2-3 incidents per year, which provides little incentive for collusion.

8 Conclusion

In this work we presented a fully decentralized model for sharing CTI. It is designed with legal and privacy requirements in mind and ensures sustainable sharing using cryptocurrency-based incentives. We implemented the DEALER platform based on the EOS blockchain and IPFS DHT and demonstrated its practical feasibility. On the platform, structured incident information is exchanged pseudonymously. Randomly selected verifiers use a set of objective CTI quality indicators to bootstrap incident reputation and help buyers select fitting incidents. Buyers and sellers are protected through dispute resolution mechanisms and exchange items based on cryptocurrency incentives.

Beyond our model and prototypical implementation, future work should conduct an in-depth security analysis considering possible attacks and their mitigations. Additionally, integration with existing incident discovery, reporting and visualization systems is essential to the platform's practical viability. For example, the incident information currently available in plaintext could be enriched by a visualization system such as the one presented by Böhm et al. [21]. Based on such integrations, the platform can be deployed on the public EOS blockchain and tested with a larger number of users. In this scenario, price discovery mechanisms and their relationship to incident data quality can be analyzed. While our infrastructure is developed with privacy in mind, future work should ensure privacy and compliance with legal requirements (i.e. GDPR) in practice.

Acknowledgment

This research was supported by the Federal Ministry of Education and Research, Germany, as part of the BMBF DINGfest project (https://dingfest.ur.de).

Compliance with ethical standards

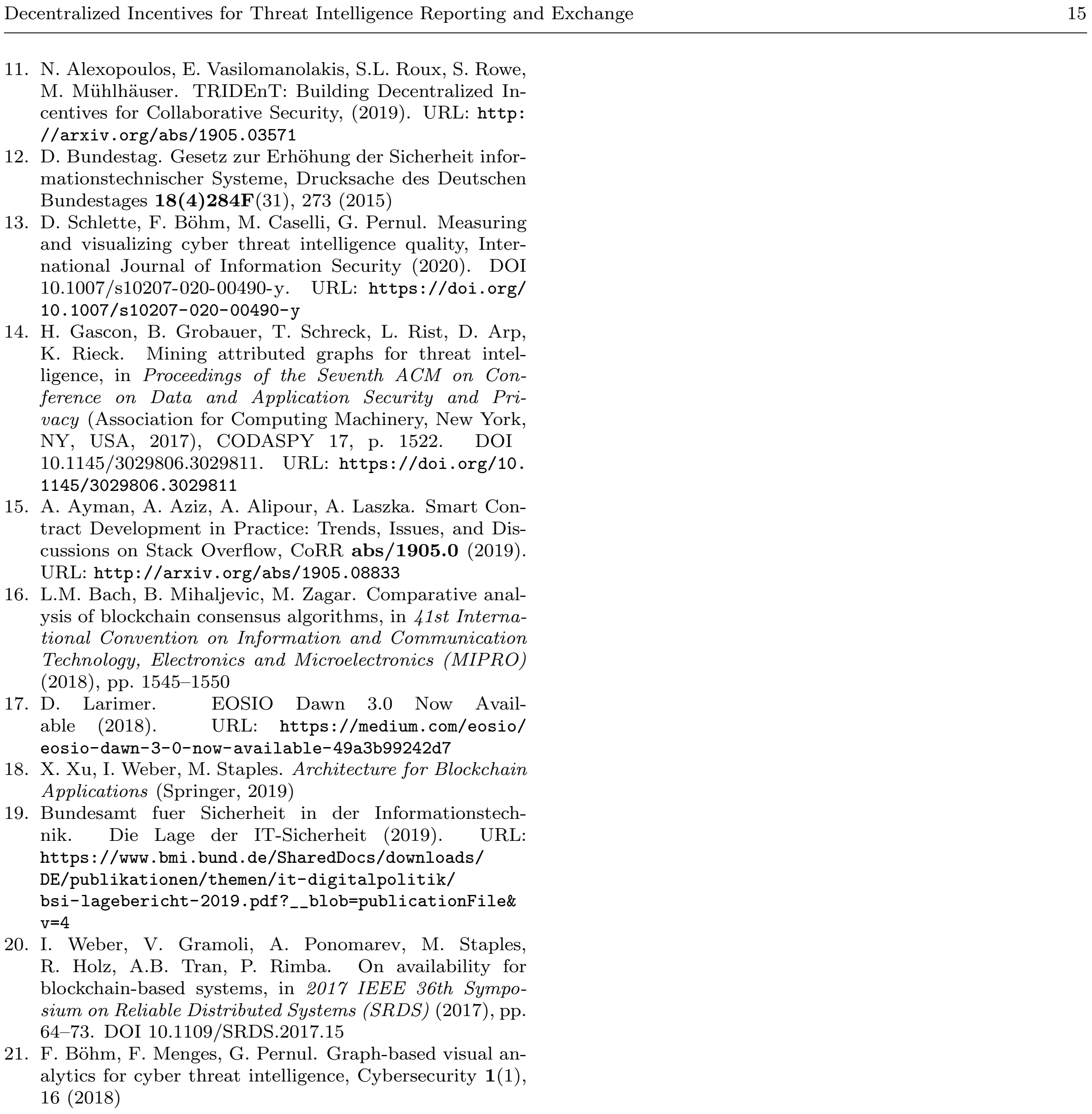
Funding Part of this research was supported by the Federal Ministry of Education and Research, Germany, as part of the BMBF DINGfest project (`https://dingfest.ur.de`).

Conflict of interest All authors declare that they have no conflict of interest.

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

References

1. IBM Corporation. X-Force Exchange. URL: `https://exchange.xforce.ibmcloud.com/`
2. Facebook Corporation. Facebook Threat Exchange (2019). URL: `https://developers.facebook.com/programs/threatexchange/`
3. C. Wagner, A. Dulaunoy, A. Iklody. MISP - The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform, Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security pp. 49–56 (2016)
4. Luatics. OPENCTI. URL: `https://www.opencti.io/en/`
5. O. Serrano, L. Dandurand, S. Brown. On the design of a cyber security data sharing system, in *Proceedings of the 2014 ACM Workshop on Information Sharing 38; Collaborative Security* (ACM, New York, NY, USA, 2014), WISCS '14, pp. 61–69
6. L. Dandurand, A. Kaplan, P. Kácha, Y. Kadobayashi, A. Kompanek, T. Lima. *Standards and tools for exchange and processing of actionable information.* November (2014)
7. S. Brown, J. Gommers, O. Serrano. From Cyber Security Information Sharing to Threat Management, Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security pp. 43–49 (2015)
8. A. Mohaisen, O. Al-Ibrahim, C. Kamhoua, K. Kwiat, L. Njilla. Rethinking information sharing for threat intelligence, HotWeb 2017 - Proceedings of the 5th ACM/IEEE Workshop on Hot Topics in Web Systems and Technologies (2017)
9. C. Sauerwein, C. Sillaber, A. Mussmann, R. Breu. Threat Intelligence Sharing Platforms : An Exploratory Study of Software Vendors and Research Perspectives, 13. Internationale Tagung Wirtschaftsinformatik, WI 2017, St. Gallen (2017)
10. C. Sillaber, C. Sauerwein, A. Mussmann, R. Breu. Data Quality Challenges and Future Research Directions in Threat Intelligence Sharing Practice, Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security pp. 65–70 (2016)



11. N. Alexopoulos, E. Vasilomanolakis, S.L. Roux, S. Rowe, M. Mühlhäuser. TRIDEnT: Building Decentralized Incentives for Collaborative Security, (2019). URL: http://arxiv.org/abs/1905.03571
12. D. Bundestag. Gesetz zur Erhöhung der Sicherheit informationstechnischer Systeme, Drucksache des Deutschen Bundestages **18(4)284F**(31), 273 (2015)
13. D. Schlette, F. Böhm, M. Caselli, G. Pernul. Measuring and visualizing cyber threat intelligence quality, International Journal of Information Security (2020). DOI 10.1007/s10207-020-00490-y. URL: https://doi.org/10.1007/s10207-020-00490-y
14. H. Gascon, B. Grobauer, T. Schreck, L. Rist, D. Arp, K. Rieck. Mining attributed graphs for threat intelligence, in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy* (Association for Computing Machinery, New York, NY, USA, 2017), CODASPY 17, p. 1522. DOI 10.1145/3029806.3029811. URL: https://doi.org/10.1145/3029806.3029811
15. A. Ayman, A. Aziz, A. Alipour, A. Laszka. Smart Contract Development in Practice: Trends, Issues, and Discussions on Stack Overflow, CoRR **abs/1905.0** (2019). URL: http://arxiv.org/abs/1905.08833
16. L.M. Bach, B. Mihaljevic, M. Zagar. Comparative analysis of blockchain consensus algorithms, in *41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (2018), pp. 1545–1550
17. D. Larimer. EOSIO Dawn 3.0 Now Available (2018). URL: https://medium.com/eosio/eosio-dawn-3-0-now-available-49a3b99242d7
18. X. Xu, I. Weber, M. Staples. *Architecture for Blockchain Applications* (Springer, 2019)
19. Bundesamt fuer Sicherheit in der Informationstechnik. Die Lage der IT-Sicherheit (2019). URL: https://www.bmi.bund.de/SharedDocs/downloads/DE/publikationen/themen/it-digitalpolitik/bsi-lagebericht-2019.pdf?__blob=publicationFile&v=4
20. I. Weber, V. Gramoli, A. Ponomarev, M. Staples, R. Holz, A.B. Tran, P. Rimba. On availability for blockchain-based systems, in *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)* (2017), pp. 64–73. DOI 10.1109/SRDS.2017.15
21. F. Böhm, F. Menges, G. Pernul. Graph-based visual analytics for cyber threat intelligence, Cybersecurity **1**(1), 16 (2018)

Bibliography

[1] Asgarli, E., and Burger, E. Semantic ontologies for cyber threat sharing standards. In *2016 IEEE Symposium on Technologies for Homeland Security (HST)* (2016), pp. 1–6.

[2] Bank of England. CBEST Intelligence-Led Testing, Understanding Cyber Threat Intelligence Operations. Tech. rep., 2016.

[3] Barnum, S. Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX^{TM}). Tech. rep., 2014.

[4] Chandel, S., Yan, M., Chen, S., Jiang, H., and Ni, T. Threat intelligence sharing community: A countermeasure against advanced persistent threat. In *2nd IEEE Conference on Multimedia Information Processing and Retrieval, MIPR 2019, San Jose, CA, USA, March 28-30* (2019), IEEE, pp. 353–359.

[5] Chantzios, T., Koloveas, P., Skiadopoulos, S., Kolokotronis, N., Tryfonopoulos, C., Bilali, V., and Kavallieros, D. The quest for the appropriate cyber-threat intelligence sharing platform. In *Proceedings of the 8th International Conference on Data Science, Technology and Applications, DATA 2019, Prague, Czech Republic, July 26-28, 2019* (2019), S. Hammoudi, C. Quix, and J. Bernardino, Eds., SciTePress, pp. 369–376.

[6] Department of Homeland Security, Cyber Infrastructure. A Guide to a Critical Infrastructure Security and Resilience (2019).

[7] Deutscher Bundestag. Gesetz zur Erhöhung der Sicherheit informationstechnischer Systeme. *Drucksache des Deutschen Bundestages 18(4)284F*, 31 (2015), 273–276.

[8] Deutscher Bundestag. Datenschutz-Anpassungs- und -Umsetzungsgesetz EU – DSAnpUG-EU . *Drucksache des Deutschen Bundestages 2017*, 44 (2017), 2097–2132.

[9] Dewald, A. Characteristic evidence, counter evidence and reconstruction problems in forensic computing. *it - Information Technology 57*, 6 (2015), 339–346.

[10] European Parliament and Council of The European Union. Directive (EU) 2016/1148. *Official Journal of the European Union 194* (2016), 1–30.

[11] FEDERICO, P., WAGNER, M., RIND, A., AMOR-AMOROS, A., MIKSCH, S., AND AIGNER, W. The role of explicit knowledge: A conceptual model of knowledge-assisted visual analytics. In *2017 IEEE Conference on Visual Analytics Science and Technology, VAST 2017, Phoenix, AZ, USA, October 3-6, 2017* (2017), B. Fisher, S. Liu, and T. Schreck, Eds., IEEE Computer Society, pp. 92–103.

[12] FENZ, S., EKELHART, A., AND WEIPPL, E. Semantic Potential of existing Security Advisory Standards. *Inproceedings of the FIRST 2008* (2008), 1–8.

[13] HEVNER, A. R., MARCH, S. T., PARK, J., AND RAM, S. Design science in information systems research. *MIS Quarterly 28*, 1 (2004), 75–105.

[14] JOINT TASK FORCE TRANSFORMATION INITIATIVE. *Guide for conducting risk assessments*. National Institute of Standards and Technology, Gaithersburg, MD, 2012.

[15] JULIADOTTER, N. V., AND CHOO, K.-K. R. Cloud attack and risk assessment taxonomy. *IEEE Cloud Computing 2*, 1 (2015), 14–20.

[16] MENGES, F., BÖHM, F., VIELBERTH, M., PUCHTA, A., TAUBMANN, B., RAKOTONDRAVONY, N., AND LATZO, T. Introducing dingfest: An architecture for next generation SIEM systems. In *Sicherheit 2018, Beiträge der 9. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI), 25.-27.4.2018, Konstanz* (2018), H. Langweg, M. Meier, B. C. Witt, and D. Reinhardt, Eds., vol. P-281 of *LNI*, Gesellschaft für Informatik e.V., pp. 257–260.

[17] MORGAN, S. 2019 Official Annual Cybercrime Report. *2019 Report by Cybersecurity Ventures sponsored by Herjavec Group* (2019), 12.

[18] ÖSTERLE, H., BECKER, J., FRANK, U., HESS, T., KARAGIANNIS, D., KRCMAR, H., LOOS, P., MERTENS, P., OBERWEIS, A., AND SINZ, E. J. Memorandum on design-oriented information systems research. *European Journal of Information Systems 20*, 1 (2011), 7–10.

[19] PEFFERS, K., TUUNANEN, T., ROTHENBERGER, M. A., AND CHATTERJEE, S. A design science research methodology for information systems research. *J. of Management Information Systems 24*, 3 (2008), 45–77.

[20] PUTZ, B., MENGES, F., AND PERNUL, G. A secure and auditable logging infrastructure based on a permissioned blockchain. *Computers & Security 87* (2019).

[21] QAMAR, S., ANWAR, Z., RAHMAN, M. A., AL-SHAER, E., AND CHU, B. Data-driven analytics for cyber-threat intelligence and information sharing. *Computers & Security 67* (2017), 35–58.

[22] SAUERWEIN, C., SILLABER, C., MUSSMANN, A., AND BREU, R. Threat intelligence sharing platforms: An exploratory study of software vendors and research

perspectives. In *Towards Thought Leadership in Digital Transformation: 13. Internationale Tagung Wirtschaftsinformatik, WI 2017, St.Gallen, Switzerland, February 12-15, 2017* (2017), J. M. Leimeister and W. Brenner, Eds.

[23] SILLABER, C., SAUERWEIN, C., MUSSMANN, A., AND BREU, R. Data quality challenges and future research directions in threat intelligence sharing practice. In *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security, WISCS 2016, Vienna, Austria, October 24 - 28, 2016* (2016), S. Katzenbeisser, E. R. Weippl, E. Blass, and F. Kerschbaum, Eds., ACM, pp. 65–70.

[24] SPRINKLE, J., RUMPE, B., VANGHELUWE, H., AND KARSAI, G. Metamodelling: State of the art and research challenges. *CoRR abs/1409.2359* (2014).

[25] STEINBERGER, J., SPEROTTO, A., GOLLING, M., AND BAIER, H. How to exchange security events? overview and evaluation of formats and protocols. In *IFIP/IEEE International Symposium on Integrated Network Management, IM 2015, Ottawa, ON, Canada, 11-15 May, 2015* (2015), R. Badonnel, J. Xiao, S. Ata, F. D. Turck, V. Groza, and C. R. P. dos Santos, Eds., IEEE, pp. 261–269.

[26] TOUNSI, W., AND RAIS, H. A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Computers & Security 72* (2018), 212–233.

[27] WAGNER, C., DULAUNOY, A., WAGENER, G., AND IKLODY, A. MISP: the design and implementation of a collaborative threat intelligence sharing platform. In *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security, WISCS 2016, Vienna, Austria, October 24 - 28, 2016* (2016), S. Katzenbeisser, E. R. Weippl, E. Blass, and F. Kerschbaum, Eds., ACM, pp. 49–56.

[28] WAGNER, T. D., MAHBUB, K., PALOMAR, E., AND ABDALLAH, A. E. Cyber threat intelligence sharing: Survey and research directions. *Computers & Security 87* (2019).