electronics

MDPI

Article

A Reference Model for Cyber Threat Intelligence (CTI) Systems

Georgios Sakellariou [1,]*, Panagiotis Fouliras [1], Ioannis Mavridis [1] and Panagiotis Sarigiannidis [2]

[1] Department of Applied Informatics, University of Macedonia, 54636 Thessaloniki, Greece; pfoul@uom.edu.gr (P.F.); mavridis@uom.edu.gr (I.M.)

[2] Department of Electrical and Computer Engineering, University of Western Macedonia, 50100 Kozani, Greece; psarigiannidis@uowm.gr

* Correspondence: geosakel@uom.edu.gr; Tel.: +30-2310-88-2850

Abstract: Cyber Threat Intelligence (CTI) is a new but promising field of information security, with many organizations investing in the development of proper tools and services and the integration of CTI related information. However, as a new field, there is a lack of a conceptual framework with corresponding definitions. This paper discusses CTI complexity factors, proposes a set of definitions of the CTI key concepts and an eight-layer CTI Reference Model as a base for CTI systems design. In addition, the proposed reference model is validated by applying it to three case studies, producing the respective CTI Reference Architectures.

Keywords: cyber threat intelligence; information security; reference model; system architecture

check for updates

Citation: Sakellariou, G.; Fouliras, P.; Mavridis, I.; Sarigiannidis, P. A Reference Model for Cyber Threat Intelligence (CTI) Systems. *Electronics* **2022**, *11*, 1401. https://doi.org/10.3390/electronics11091401

Academic Editor: Paulo Ferreira

Received: 5 April 2022
Accepted: 20 April 2022
Published: 27 April 2022

1. Introduction

Cyberspace is widely known as an insecure environment, with the leading cybersecurity vendors reporting an approximately 31% increase in the average number of attacks that a company phase [1]. In addition, ransomware denial of service (RDoS), crypto-jacking, disinformation, info stealers, and mobile malware are new or most frequent sources of incidents an organization faces [2].

The overall risky cyberspace environment has led organizations to invest up to USD 150 billion for cybersecurity in 2021 alone (Gartner Forecasts). However, organizations still have limited security resources. Consequently, Cyber Threat Intelligence (CTI) is suggested so that the attacker's motivation and tactics [3] can be better understood. Investment in CTI is set at sixth place among seventeen cybersecurity components by the most significant increase in cost [4]. Furthermore, CTI production and sharing, in particular, can play a vital role as a means of collaboration among organizations' leaders [4] or, as in the case of the European Union, which announced the foundation of the Joint Cyber Unit [5] on 2019 aiming at real-time information exchange, among nation-states. Additionally, the role of CTI is considered critical for zero-day attack prevention [6].

Unfortunately, several factors such as data volume, variety of data sources, descriptions and formats [7], and data inconsistency [8] increase an organization's CTI implementation complexity and highlight the importance of automation systems that support CTI analysts. Requirements for such systems, also known as Threat Intelligence Platforms (TIPs), have already described in detail [9]. More importantly, the lack of a CTI reference model has led to systems either partially meeting TIPs requirements, addressing part of the factors affecting the CTI [10], or missing essential phases of the Intelligence Cycle [11]. Methodologies for selecting and evaluating such systems have been proposed [12,13]; however, they face the problem from the consumer's point of view rather from the system designer's.

As stated in [14], the modeling of complex domains and systems is used in research for simplification and representation purposes. Unfortunately, the meaning of the terms *model*, *framework*, *architecture*, *reference model*, *reference architecture*, and *system architecture* are not consistent in the bibliography [15–17]. Although the terms *reference model* and *reference architecture* are ambiguous, it is stated that a reference model's objective is *"to streamline the design of enterprise- individual models by providing a generic system"* [18]. In [14], the authors define the reference model as *"a generic abstract representation for understanding the entities and the significant relationships among those entities of some area"*. They also propose that a reference architecture be *"an abstract description of a specific system"*, which *"aims at structuring the design architectures for a given domain by defining a unified terminology, describing the functionality and roles of components, providing template components, providing example architectures, and defining a development methodology"*.

The motivation for this paper was the lack of a CTI reference model for the systematic development of a CTI system combined with the absence of holistic research on those complexity factors that affect the design of such systems. The main contributions are:

- The aggregation and classification of the complexity factors that affect CTI and the design of CTI systems.
- A set of definitions for CTI key concepts.
- The development of an eight-layer CTI reference model.
- A systematic requirements analysis method for the design of CTI reference architectures.

The rest of this paper is organized as follows. Section 2 presents a set of CTI definitions. Next, Section 3 presents the reference model development methodology. A CTI reference model is proposed in Section 4, followed by its validation in Section 5. Finally, in Section 6, we present our conclusions and future work.

2. Definitions of Key Consepts

The lack of precise definitions of key concepts of CTI leads in many cases to misunderstanding and confusion when authors use different terms for the same concept. This led us to propose a set of basic definitions used in the rest of this work. Specifically, we define the terms *threat intelligence process, CTI source, CTI product, CTI producer, CTI consumer,* and *CTI system* as follows:

- A ***threat intelligenceprocess*** is any process consisting of those actions taken by the security analyst to transform raw data into usable information.
- A ***CTI source*** is any data source that can contribute to the situational awareness of defense capabilities against cyber threats.
- A ***CTI product*** is the outcome of any threat intelligence process meeting a set of predefined quality characteristics.
- A ***CTI producer*** is any entity that applies a threat intelligence process to produce CTI products.
- A ***CTI consumer*** is any entity able to use CTI products to increase its defense capabilities or take decisions about issues relevant to cybersecurity.
- A ***CTI system*** is any cybersecurity system, tool, or system capable of performing or supporting part of or all the actions of a threat intelligence process.

After examining various CTI definition proposals [19,20], we concluded that none of them was suitable. Hence, we proceeded with the proposal of a new, generic CTI definition, as follows:

Cyber Threat Intelligence is the cybersecurity domain in which data collected from various CTI sources are analyzed and assessed regarding threat actors and their motivation, methodology of a cyber-attack, and victim(s) to support defenders towards detecting, preventing, or (ideally) predicting a cyber-attack and making appropriate decision(s).

Figure 1 presents the relationships between the CTI key concepts mentioned above.

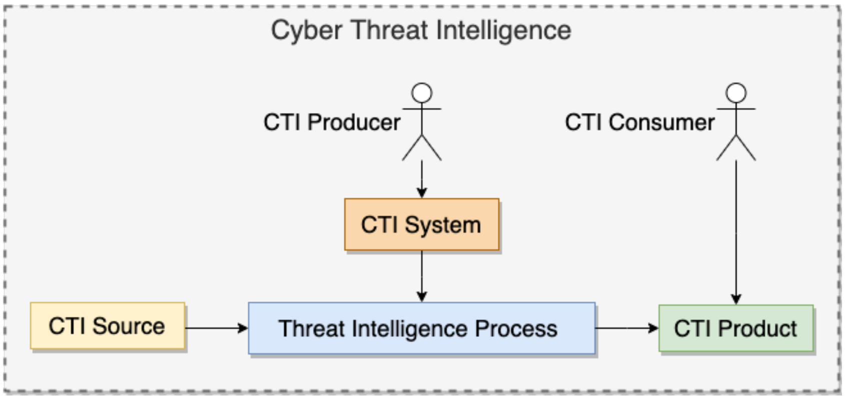


Figure 1. CTI Key Concepts relationships.

3. Methodology

Following a design methodology is crucial for the development of a reference model. The authors of [21] propose a four-stage methodology: problem definition, construction of a frame of reference, core construction, and validation. The authors of [22] follow a similar approach, distinguishing four stages: scope definition, business process modeling, process merging, and reference model verification. Both methodologies include similar activities for developing a reference model. Nevertheless, as stated in [21], in new research areas, where no widely accepted definitions and reference models exist, a reference model can be designed via analysis and abstraction of the elements and practices proposed in the bibliography.

Before we continue with the presentation of the methodology, we explain the terms used in this article to increase its readability. Specifically, we use the term *CTI concepts* to refer to general ideas, procedures, and perspectives related to CTI. Next, we use the term *CTI concepts category* to refer to a set of CTI concepts that describe or deal with similar subjects of CTI. Finally, we use the term *model element* [23,24] to describe the basic building blocks of the reference model, which expresses a structural or behavioral feature of a system.

Based on the above, we follow a four-phase design methodology in this paper, as depicted in Figure 2. This is the foundation for developing our proposed *CTI reference model*.

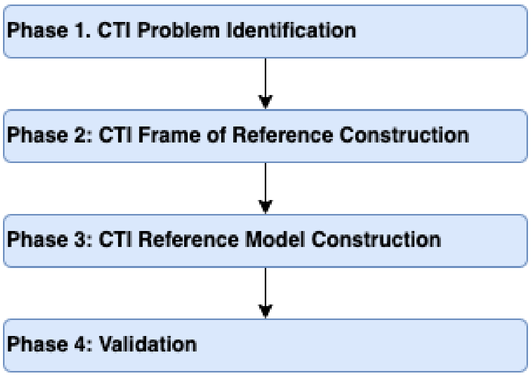

Phase 1. CTI Problem Identification
Phase 2: CTI Frame of Reference Construction
Phase 3: CTI Reference Model Construction
Phase 4: Validation

Figure 2. CTI Reference Model Design Methodology.

During the first phase, we identify the CTI problem. In the second phase, we construct a CTI frame of reference by reviewing and analyzing the existing domain knowledge and identifying the model elements. In general, a frame of reference is a set of ideas, conditions, or assumptions that determine how something will be approached, perceived, or understood [25]. The CTI reference model is constructed in the third phase. Finally, in the fourth phase, the CTI reference model is validated [21] by applying it against a set of case studies.

Next, we analyze each phase, in detail, adducing best practices and various approaches found in the bibliography.

3.1. CTI Problem Identification

As stated in [22], a reference model should: (i) represent best practices, (ii) be widely acceptable, and (iii) be reusable. For this reason, we need to review the CTI related standards and bibliography to discover potential commonly accepted definitions of crucial CTI concepts, which could be subsequently adopted during the CTI reference model construction.

Consequently, we begin with the identification of the various aspects of the CTI problem, focusing on CTI concepts and especially on those that increase CTI system complexity (i.e., CTI complexity factors). We conclude phase 1 with a generic definition of the CTI problem.

3.2. CTI Frame of Reference Construction

A broad understanding of the CTI domain is a prerequisite, for constructing a reference model that represents a logical and innovative outcome for cybersecurity. There are two steps in this phase.

In the first step, we analyze the bibliography review results from the previous phase while enriching them with the analysis of existing CTI systems. Our purpose is to identify and categorize the model elements related to the characteristics of CTI systems.

In the second step, we construct a frame of reference based on these model elements by applying a set of structuring criteria [21], which we define for the CTI problem. The structuring criteria must be clear, intelligible, and comprehensive. Following [21], such criteria are the management levels differentiation, and a processes breakdown. As a result, the *frame of reference* depicts a high-level perspective of the CTI problem, which comprises the model elements and their potential relationships. This assists in structuring all knowledge related to CTI, while simultaneously acting as a guide for *CTI reference model* construction.

3.3. CTI Reference Model Construction

This phase represents the core of the construction process, which results in the *CTI reference model*. During this phase, we combine the CTI frame of reference with CTI scenarios (abstract use cases, where CTI systems are used by security analysts), while simultaneously utilizing existing CTI systems' architectures to identify the layers that comprise a CTI reference model.

3.4. Validation

As the authors of [21] suggest, this phase seeks to validate the resulting reference model by solving real problems. To this objective, we follow a case study-based approach as proposed in [26]. Specifically, we define a set of case studies derived from actual incidents. Then, we use the *CTI reference model* to analyze those case studies and produce conformant *reference architectures* of CTI systems, thus concluding its validation.

4. CTI Problem Identification

4.1. CTI Problem Identification

Studying the bibliography related to CTI, we observe the existence of heterogeneous analyses of CTI concepts. For instance, information sharing and data analysis are two of the most widely analyzed concepts of CTI compared to the rest.

In general, however, we identify three broad categories of CTI concepts, which help us towards analyzing them, defining the generic problem of CTI, and finally developing a reference model:

1. *CTI intelligence views*
2. *CTI intelligence cycle*
3. *CTI complexity factors*

4.1.1. CTI Intelligence Views

CTI can have a multilevel contribution to an organization's security. Authors have proposed different points of view to standardize and categorize the functions and the role of CTI within an organization.

H. Dalziel [27] focuses on building CTI capability, with the organization's individual goal playing a central role. He also states that intelligence, in general, is the usable output of a logical and analytical process; no other distinction exists except the quality of information measured in relevance, actionability, and usability. In [28], the authors present different approaches of what CTI is by dividing it into four subtypes (strategic, operational, tactical, and technical), which are also adopted by [29]. The human factor is at the core of CTI in [30], with products and CTI being classified, in a more general sense, either as formal or informal, mainly based on how a security analyst perceives the source of CTI. In [31], the author distinguishes two categories of CTI based on whether a security analyst is involved. *Operational CTI* is responsible for identifying, collecting, processing, and enriching raw data. By contrast, *strategic CTI* defines the threat's impact, sustains valuable information sources, and defines adversaries' attribution and trends.

Finally, the authors of [32] consider CTI to have a twofold nature, and, by extension, they distinguish two CTI categories: *tactical and strategic*. Tactical CTI is low-level intelligence (e.g., log files), whereas strategic CTI is high-level intelligence (e.g., security analysts' opinions). The aforementioned categorizations of CTI do not affect the actual CTI work of a security analyst, but do affect the goals of CTI and the interpretation of its results. So even though the division of CTI proposed in [28] is the most widely acceptable, we should consider all proposals during the development of a *CTI reference model*. Therefore, we introduce the generic term CTI intelligence view to refer to them for clarity and consistency.

4.1.2. CTI Intelligence Cycle

The CTI intelligence cycle represents the successive phases during intelligence product creation and utilization, independent of intelligence types such as Human Intelligence (HUMINT) and Signals Intelligence (SIGINT) [33], expressing the different works an analyst has to be involved in. Nevertheless, it is important to note that the CTI intelligence cycle should not be confused with other cybersecurity models or methodologies, such as the diamond model [34] and the cyber kill chain [35]. Both are linked with the CTI intelligence cycle products and not the actual processes.

The are several approaches about the number and scope of the CTI intelligence cycle's phases in the bibliography. For example, in [36], a single, four-phase intelligence cycle was proposed consisting of the: direction, collection, processing, and dissemination phases. However, the complexity of intelligence work led many researchers to reconsider the number of phases and to propose intelligence cycles with six (planning and direction, collection, processing and exploitation, analysis and production, dissemination and integration) [11] or eight phases (direction, collection, evaluation, analysis, integration, interpretation, dissemination) [37,38], or even propose a different approach altogether, such as the intelligence web [39].

Although the concept of intelligence cycle is usually the basis of planning cyber intelligence operations, few CTI-specific publications deal with the CTI intelligence cycle itself. For instance, in [28], the authors propose a CTI intelligence cycle comprising requirements, collection, analysis, production, and evaluation phases. Additionally, in [9], the TIPs requirements study is based on a five-phase CTI intelligence cycle (planning and direction, collection, processing and exploitation, analysis and production, dissemination, and integration).

Regardless of the proposed cycle, it is evident that the phases describe the various works of a security analyst during the production of CTI products. Hence, to develop a reference model for the CTI systems that support the security analysts it is essential to follow a specific intelligence cycle, which is the most generic and the most acceptable. Since most of the existing papers adopt the six-phase *CTI intelligence cycle* of [11] (planning and direction, collection, processing and exploitation, analysis and production, dissemination and integration), we use it for the rest of this work.

4.1.3. Complexity Classes

In the bibliography, the authors present many factors that may increase the difficulty of a CTI system to help a security analyst creates CTI products. In order to describe and better analyze all these factors, we introduce the term *CTI complexity factors*, and classify them into five main *classes*. A class may contain either single factors or (one or more) groups of factors. Our classification into classes and groups is based on the empirical study of complexity factor characteristics (e.g., does the factor have a technical impact to the CTI?). Of course, as CTI continues to grow, new factors are expected to emerge. The five **classes of CTI complexity factors** are:

1. CTI Information Sharing
2. CTI Security Operations
3. CTI Big Data
4. CTI Representation
5. CTI Quality of Intelligence

CTI Information Sharing

The *CTI information sharing* class includes the complexity factors related to the exchange of CTI between different entities [28,40]. In the bibliography, those factors range from purely theoretical to technical ones. For systemization purposes, we grouped them as follows (see Figure 3):

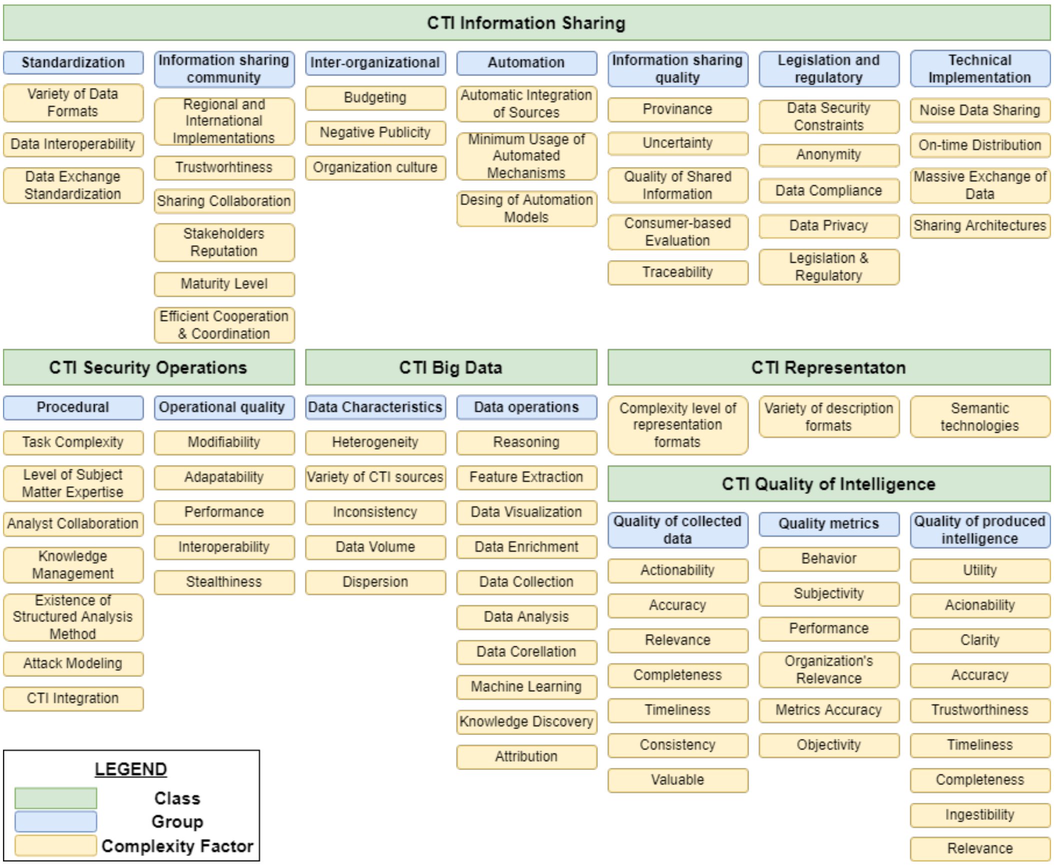


Figure 3. CTI Complexity Factors and their Classification.

- *Legislation and regulatory factors*. These complexity factors are the information sharing legislative and regulatory framework [29,33,41], i.e., sharing CTI products under

GDPR [42], the data privacy of shared data [43,44], data compliance [43], anonymity [44], and data security constraints [43].

- *Standardization factors*. These are the lack of data exchange standardization [41,43], the use of various data formats [45], and data interoperability [44].
- *Inter-organizational factors*. This group includes the factors: technology integration into organizations [41], organization culture [29], budgeting [29], and negative publicity [29].
- *Information Sharing community factors*. This group includes the factors: regional and international implementations of information sharing [41], maturity level [32], trustworthiness [29], stakeholders' reputation [44], efficient cooperation and coordination [41], and collaboration [44] of information sharing community members.
- *Information sharing quality factors*. This group includes factors related to the *CTI quality of intelligence class*, but specialized in information sharing. These factors are: consumer-based evaluation of intelligence [32], quality of shared information [29], traceability and provenance of threat intelligence [43], and uncertainty of sharing [46].
- *Automation factors*. These factors are: minimum usage of automated mechanisms [33,43], and design of CTI automated sharing [44].
- *Technical implementation factors*. This group includes factors that, in some cases, are also related to other classes such as CTI big data; these factors are: on-time distribution of relevant threat intelligence products [43], noise data sharing [43], and massive exchange of data [9]. Note that the factors of sharing architectures also belong to this group [47].

CTI Security Operations

The *CTI security operations* class comprises factors that add complexity due to the relation of CTI with the security operations. We further classify them into two separate groups:

- *Security operations procedural factors*. These factors are: task complexity, level of subject matter expertise, analysts' collaboration, existence of structured analysis method [48], organizational knowledge management [49], method of attack modeling [50], and level of CTI integration [32].
- *Quality factors of security operations*. This group contains: performance, interoperability, adaptability, modifiability, and stealthiness [51].

CTI Big Data

The *CTI big data* class includes intelligence data handling and processing factors. We assort them into two groups:

- *Data operations factors*. This group contains the factors: data collection [9,52,53], data analysis [8], data visualization [8], reasoning [46], knowledge discovery [46,52], attribution [46], data enrichment [54], feature extraction [52,54], data correlation [55], and application of machine learning (e.g., model construction and validation) [52].
- *Data characteristics factors*. This group includes factors related to the characteristics of intelligence data. These factors are: data volume [8,32], data inconsistency [8], variety of intelligence data sources [56], dispersion of data [56], and the heterogeneity of data [53].

CTI Representation

The *CTI representation* class deals with the factors involved in intelligence products representation. These factors are: complexity level of representation formats [57], variety of existing intelligence description formats [45,58], and the use of semantic technologies (e.g., ontologies) [59].

CTI Quality of Intelligence

The CTI quality of intelligence class includes all factors related to quality in CTI, which is considered one of the fundamental CTI challenges [9]. We classify those factors into three groups:

- *Quality metrics factors*. According to [31], quality metrics are considered essential in CTI. This group includes factors related to the measurement of intelligence quality. These factors are objectivity, subjectivity, performance, behavior, accuracy of metrics [60], and organization's relevance [40] of produced intelligence.
- *Quality factors of collected data*. This group's factors are related to the quality of the data collected to be processed for intelligence purposes. These factors are: collected data accuracy (e.g., dates, incident type, contact details) [61], timeliness [61], completeness, [61], consistency [61], relevance [27], actionability [27], and value [27].
- *Quality factors of produced intelligence*. The group includes factors related to the quality characteristics of the CTI products. These factors are: the accuracy [32,33,44,62,63], clarity, [62], utility of the products [62], relevance [32,43,44,63], timeliness [32,33,43,44,63], actionability [32,33,44], completeness [33,44,63], ingestibility [44] and trustworthiness [44] of threat intelligence.

4.1.4. CTI Related Standards

A security analyst has to deal with various standards to support intelligence collection, analysis, description, and dissemination. A CTI system should integrate those standards for compatibility with the rest of the cybersecurity industry. However, we do not focus on analyzing these standards in this work; we focus, instead, on identifying whether these standards provide widely accepted definitions of CTI concepts.

After a thorough study, we determined that most CTI related standards (see Table 1) do not define CTI concepts useful for the development of our CTI reference model. Instead, they standardize specific procedures related to CTI (e.g., CTI information sharing).

Table 1. CTI Related Standards.

CTI Related Standards	Description
YARA [64]	It provides the means of malware description, identification and classification.
CWE [65]	A language and a list of software and hardware weaknesses.
CVE [66]	A language (catalog) for identified and defined cybersecurity vulnerabilities.
CCE [67]	It provides identifiers for common configuration issues.
CPE [68]	A language and dictionary for information systems, software, and packages naming.
MAEC [69]	Malware attributes enumeration and characterization provides a structured way to describe a malware.
CAPEC [70]	A dictionary and a hierarchy of common attack patterns.
ATT&CK [71]	A knowledge base and a common language for attack tactics and techniques.
Cyber Kill Chain [35]	A framework that models the adversary activities to succeed his objectives.
CybOX [72]	A common language for the description of cyber observable.
STIX [73]	A CTI information exchange language and serialization format.
Diamond Model [34]	It provides an intrusion analysis approach and methodology.
OpenIOC [74]	It provides a standard for the description of artifacts during an investigation.
TLP [75]	A protocol ensuring the information sharing of sensitive data.
TAXII [76]	A CTI information exchange protocol and standard.
IODEF [77]	A framework for data representation of cyber security incidents.
VERIS [78]	A common language for describing security incidents.

After thoroughly studying them, we concluded that most standards do not provide definitions of key concepts. Instead, they focus on the standardization of specific procedures related to CTI, e.g., information description.

4.1.5. CTI Problem Definition

To conclude the first phase of the methodology, we define the generic CTI problem. The CTI problem describes the actual reason behind developing a CTI system and answers the question of "what does a CTI system deal with?". Therefore, we define the CTI problem as: *the challenge of transforming and using any information gathered from various CTI sources into CTI products*.

4.2. CTI Frame of Reference Construction

4.2.1. Model Elements Identification

Recall from Section 4.1 (CTI Problem Identification) that we classified CTI concepts in three broad categories, namely: CTI intelligence views, CTI intelligence cycle, and CTI complexity factors. Therefore, to effectively design a frame of reference, we need first to identify the model elements [23] deriving from those three categories of CTI concepts. Consequently, we list the model elements based on these three categories and validate them by identifying them in existing CTI systems (we present only overall statistics for those solutions due to confidentiality agreements that do not allow us to directly refer on the characteristics of the private solutions). More specifically, we studied the capabilities and functionalities of twenty-three private and open-source threat intelligence systems. For most of the proprietary systems, we also interviewed their technical representatives. Figure 4 presents the identified model elements, arranged in three categories of CTI concepts.

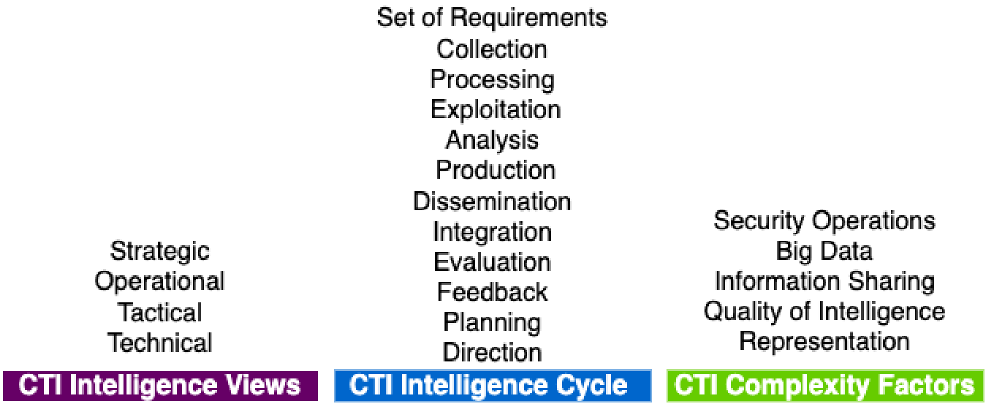


Figure 4. Identified Model Elements arranged in three CTI Concepts Categories.

In all cases, we identify a model element in a CTI system, only if the CTI system includes at least one functionality related to the particular model element. For example, we identify the Collection model element in a CTI system, only if this CTI system can collect data from CTI sources. We consider that a system has a functionality if it can do the job described by the functionality [79].

The only exception arises for the five model elements related to the category of CTI complexity factors. In this case, we identify a model element in a CTI system, if the latter includes a functionality handling at least one complexity factor of the class to which the model element is related (cf. Figure 3). For example, suppose a CTI system measures the actionability of the collected data. According to Figure 3, this factor belongs to the Quality of Intelligence class of CTI complexity factors. Therefore, we conclude that the Quality of Intelligence element does exist.

In Figure 5, we present each model element's percentage rate of occurrence within the set of existing CTI systems, up to the writing of this paper. Even though the rate of model elements occurrence varies, the important fact is that all model elements appear in CTI systems.

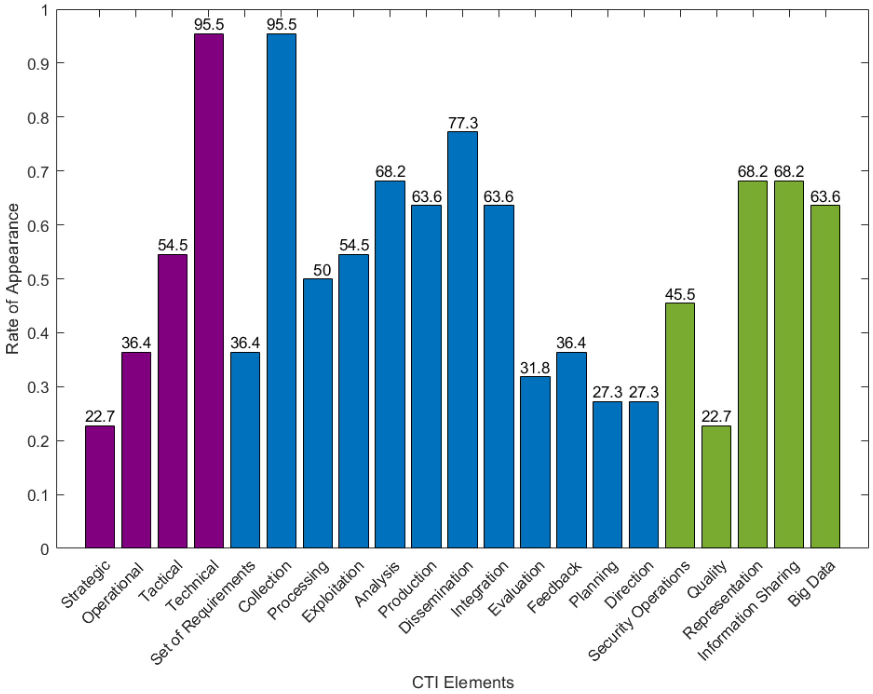


Figure 5. Rate of CTI Elements' Appearance in existing CTI Systems.

4.2.2. CTI Frame Reference Construction

The identified model elements are the building blocks of a CTI frame of reference because they abstractly represent all CTI-related concepts that may affect the work of a security analyst during a threat intelligence process. In this section, we present the structuring criteria used for the construction of the proposed frame of reference. Since no related work exists, we propose those criteria, empirically, based on our extensive survey of the CTI domain:

- ***Criterion 1***: *Separation of model elements from the CTI intelligence cycle into managerial and practical*. We consider as practical the model elements that play a part in data processing, and as managerial the model elements related to the governance of the CTI intelligence cycle. Criterion 1 allows us to distinguish between those model elements that a CTI system can implement and those that it cannot (since they constitute the management framework of CTI).
- ***Criterion 2***: *Time-based division of model elements from CTI intelligence views into long- and short-term*. According to the bibliography [28,29,31,32], CTI intelligence views affect both the kind and the lifetime of CTI products. Therefore, criterion 2 allows us to distinguish model elements of the CTI intelligence views class according to their effect on CTI products' ephemerality.
- ***Criterion 3:*** *Origin-based division of model elements from CTI complexity factors into internal and external*. We consider such model elements as either internal (emanating from CTI itself), or external (imposed externally on CTI), because a CTI reference model (at a minimum) should be able to deal with internal complexity factors.
- ***Criterion 4:*** *Identification of unique processes*. Specifically, we identify model elements corresponding to unique processes, typically undertaken by a security analyst.
- ***Criterion 5:*** *Identification of relation paths between model elements representing a unique process*. This criterion identifies the relation paths connecting unique processes in a logical sequence, which, when implemented by a CTI system, can produce CTI Products.

By applying criteria 1–3 on model elements, we arrange model elements as depicted in Figure 6.

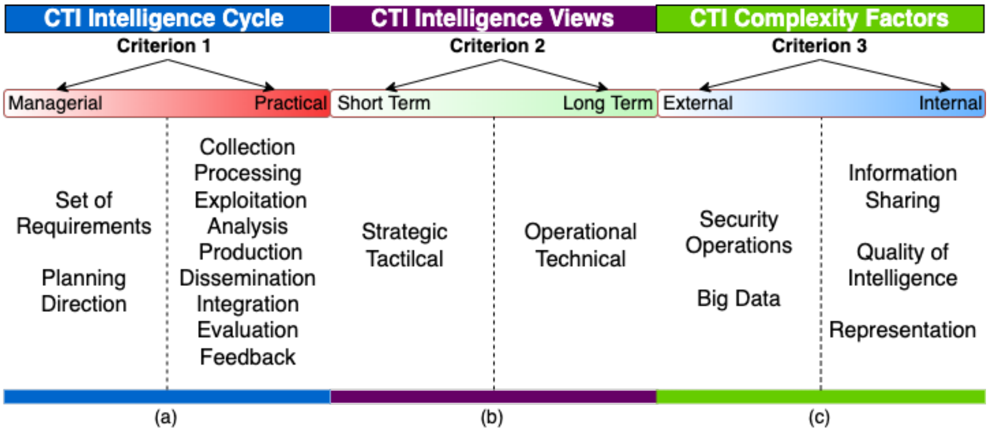


Figure 6. Applying: (**a**) Criterion 1, (**b**) Criterion 2, and (**c**) Criterion 3 to model elements.

Applying criterion 4 on model elements related to the first CTI concepts category (i.e., CTI intelligence views, cf. Figure 7a), we cannot identify any related unique process, directly. Nevertheless, those elements (strategic, operational, tactical, technical) imply the different types of requirements, and by extension, the different kinds of CTI products that can affect a CTI system's design. For example, a CTI system designed to fulfill technical view requirements is expected to handle low-level technical information (e.g., log files). Next, we apply criterion 4 on the model elements related to the second CTI concepts category (i.e., CTI intelligence cycle, cf. Figure 7a). As a result, we deduce that each element represents a unique process. Applying criterion 4 on model elements related to the third CTI concepts category (i.e., CTI complexity factors, cf. Figure 7a), we observe that they do not correspond to unique processes, although they may affect them.

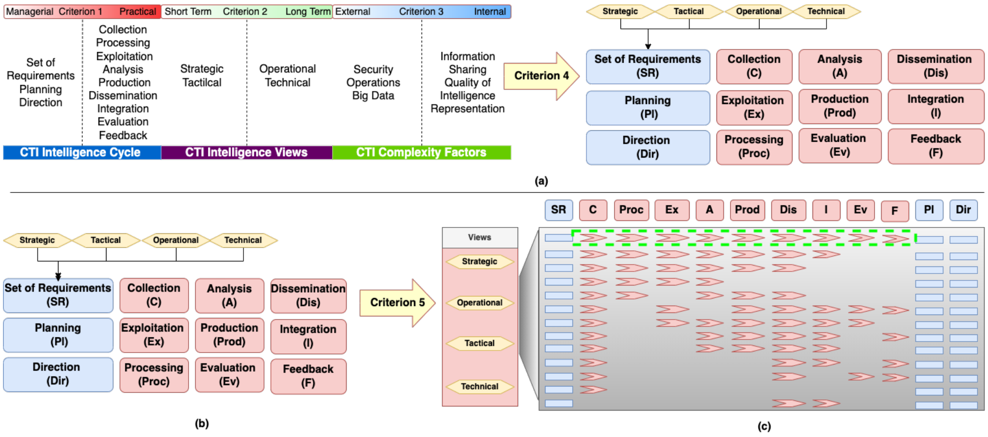


Figure 7. Application of: (**a**) Criteria 4, (**b**) and (**c**) Criteria 5.

We apply criterion 5 (cf. Figure 7b) to identify the relation paths connecting the unique processes identified earlier by the application of Criterion 4. Figure 7c depicts the relation paths; each relation path reveals a possible way that processes can be combined to create CTI products. Those paths were identified in both CTI systems proposed in the bibliography, as well as in existing CTI systems (cf. Figure 8) after analyzing their functionality. A relation path, such as SR-C-Proc-Ex-A-Prod-Pi-Dir (Set of Requirements-Collection-Exploitation-Analysis-Production-Planning-Direction), is identified in an existing CTI system, provided that both of the following conditions are met:

- The model elements comprising the relation path can be identified in it (e.g., a collection module exists in a CTI system).
- The CTI system can produce CTI products by combining their functionality following this relation path.

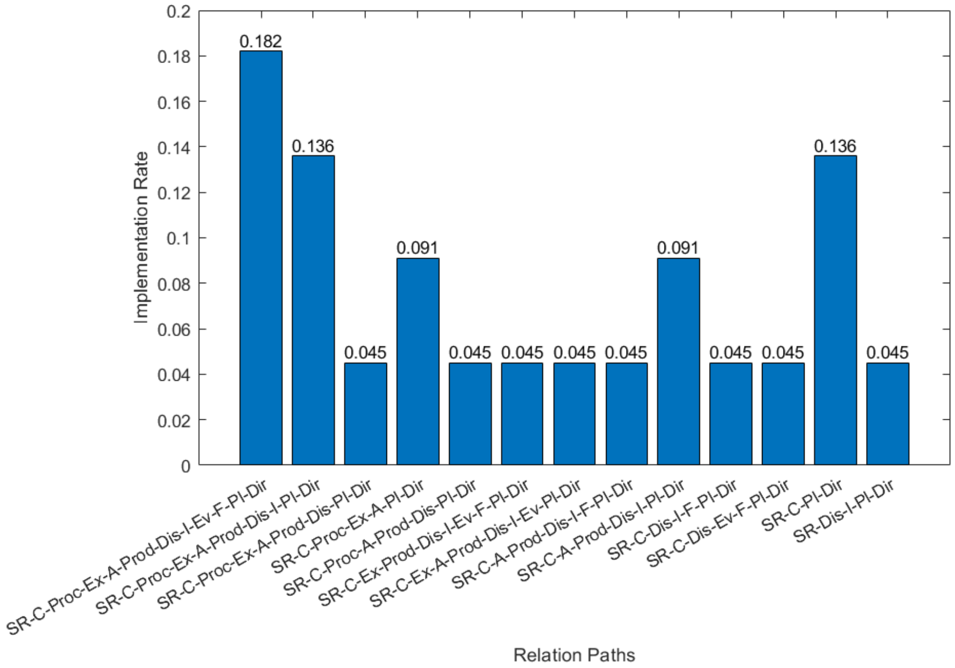


Figure 8. Implementation Rate for each Relation Path in existing CTI Systems.

Finally, we construct the CTI frame of reference, using the relation path with the maximum number of processes (cf., Figure 7c, within green dashed rectangle), because a frame of reference needs to be as general and as abstract as possible. Figure 9 depicts the resulting CTI frame of reference with nine layers (from Collection to Feedback). This provides the basic structure for a CTI reference model.

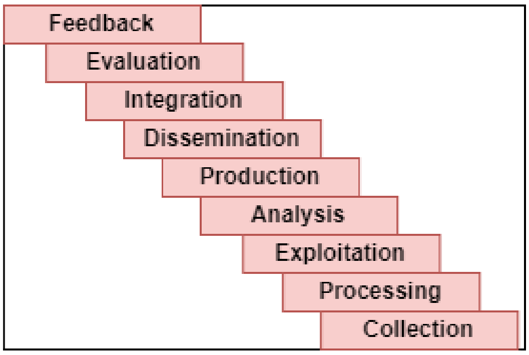


Figure 9. CTI Frame of Reference.

4.3. CTI Reference Model Construction

4.3.1. Complexity Factors concerning CTI Frame of Reference Layers

In general, the CTI complexity factors affect all recognized model elements and, by extension, the CTI frame of reference. However, a CTI reference model should at least determine those complexity factors that may affect both a CTI system's functionality, as well as its design. Consequently, we focus our attention to the CTI complexity factors which can be handled by one or more dedicated modules in a CTI system architecture (e.g., an anonymizer module that handles the complexity factor of Anonymization).

Next, for each CTI complexity factor we examine whether it belongs to any of the layers of the CTI frame of reference; if not we remove the particular CTI complexity factor since does not affect a system's design. For example, in the class of CTI information sharing (within the group of information sharing community factors), the regional and international implementations (see Figure 2) factor may affect the volume of collected data or the regulatory framework of a CTI analyst's work. However, no specific module integrated into the design of a CTI system could ever handle that. On the other hand, it is possible to have a module handling the stakeholders' reputation factor (e.g., a ranking subsystem); hence, the particular CTI complexity factor belongs to the dissemination layer of the CTI frame of reference. Table 2 depicts the CTI complexity factors categories concerning CTI frame of reference, while a more detailed depiction of the CTI complexity factors and their relationships with the CTI frame of reference appears in Figure S1.

Table 2. CTI complexity factors categories concerning CTI frame of reference.

	Collection	**Processing**	**Exploitation**	**Analysis**	Production	**Dissemination**	**Integration**	**Evaluation**	**Feedback**
CTI Information Sharing					■	■	■	■	■
CTI Security Operations	■	■	■	■	■	■	■	■	■
CTI Big Data	■	■	■	■					
CTI Representation				■	■				
CTI Quality of Intelligence					■			■	■

4.3.2. CTI Scenarios

As already mentioned, we studied the capabilities of twenty-three CTI systems. In combination with the bibliography on CTI products, we identified the following CTI scenarios:

1. Collect raw data and produce CTI products.
2. Use of CTI products to create new or enrich existing CTI products.
3. Use of CTI products as feed-in defense mechanisms.
4. Use of CTI products to produce no CTI products.

4.3.3. CTI Reference Model

The construction of the *CTI reference model* combines the CTI frame of reference (Section 4.2.2), the relation of complexity factors with the CTI frame of reference (Section 4.3.1), and the CTI scenarios (Section 4.3.2) [14,80]. We use the term *layer* to express a component that offers a critical capability to a security analyst. The term *function* represents a component service that addresses one or more complexity factors or specifies the offered capability of a layer. Information follows a bottom-up direction, indicating the relationships between layers.

The proposed *CTI reference model* consists of eight layers (cf. Figure 10). The first (lowest) is the ***Selection*** layer which adds to a CTI system the capability to deal with disparate CTI sources; as CTI scenarios indicate, such sources spread across a broad spectrum, ranging from logs, news feeds, and other raw data to CTI products like STIX artifacts.

We identify two primary functions in this layer: *CTI Products Selection* and *Raw Data Selection*. These functions handle the CTI complexity factors related to CTI system evaluation, categorization, and selection of CTI sources.

Moreover, this layer includes the functions of *Traceability, Trustworthiness,* and *Stealthiness*. The first two functions can be applied to CTI sources that provide CTI products, whereas the latter can be applied to CTI sources which provide external raw data (data not owned by the owner of CTI system). More specifically, the *Traceability* function provides a CTI system with the capability to trace the creator and modifiers of each CTI product; the *Trustworthiness* function deals with trust matters between the designed CTI system and the CTI source; finally, *Stealthiness* deals with any issues that can reveal the identity of the CTI system during the selection of a CTI source.

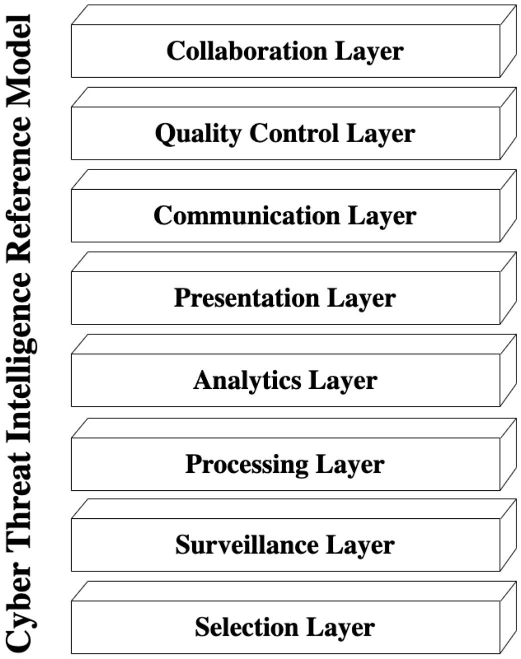


Figure 10. CTI Reference Model.

The ***Surveillance*** layer deals with data collection capability from the selected CTI sources. There are two primary functions: *Automatic Data Collection* and *Manual Data Collection*, which address the complexity factors related to the CTI frame of reference collection layer. Additionally, we can identify the *Large Volume of Data Collection* function referring to the capability of big data handling of a CTI system.

The ***Processing*** layer refers to the necessary capabilities of a CTI system to prepare the collected data for analysis. Such capabilities are data attribution, feature extraction, correlation, and normalization. Here, we identify two functions: *Data Aggregation* and *Data Enrichment*. The first function is the capacity of a CTI system to handle data gathered from different CTI sources (thus addressing complexity factors such as data inconsistency, heterogeneity of data, and variety of CTI product formats). The second function refers to the ability of a CTI system to combine raw data or CTI products from various CTI sources when they all refer to the same subject.

The ***Analytics*** layer comprises all the capabilities used by a CTI system to transform the processed data into exploitable CTI information. We identify three functions: *Manual Analysis, Attack Modeling*, and *Knowledge Discovery*. The first function is the capacity of a CTI system to allow subject-matter experts to affect data analysis. The second is the ability of a CTI system to support the use of an attack modeling methodology (e.g., Diamond Model [34]). Finally, the third function handles the analytics of big data and refers to the capacity of a CTI system to apply machine learning, reasoning, and semantic analysis methods to the processed data.

The ***Presentation*** layer is the capability of a CTI system to either transform the results of Analytics in CTI products or transform and present them to the analysts. Transformation in this layer means of structured or unstructured formatting. There are two functions in this layer that address specific complexity factors: *Visualization* and *Anonymization*. The first function is the CTI system's capacity to visualize the Analytics layer's results or the final CTI products. In addition, when a CTI system has the second function, an analyst can implement anonymization techniques in CTI products before they become available outside the organization.

The ***Communication*** layer refers to the capability of a CTI system to effectively communicate the created CTI products to other systems. This layer addresses the interoperability and communication issues of a CTI system. We can identify two functions that handle complexity factors related to this layer: *CTI Products Exchange* and *Privacy Protection*. The first function is the capacity of the CTI system to act as a member of a CTI information-sharing community, addressing the complexity factors of automation of sharing, and traceability and provenance of the created CTI products. The second function is the capacity of a CTI system to protect and prevent the sharing of CTI products that incorporate the CTI system owner's private data with entities not explicitly allowed by the owner.

The ***Quality Control*** layer provides a CTI system with the capability to manage quality issues like its efficiency, the performance of the analysts, and the quality characteristics of CTI products. In addition, we identify three functions that address complexity factors related to this layer: *Feedback Collection*, *Quality Metrics Calculation*, and *CTI Products Evaluation*, with the last two functions interrelated. The first is the capacity of a CTI system to collect feedback about the created CTI products. The second is the capacity of a CTI system to calculate quality metrics related to CTI product quality characteristics such as accuracy, actionability, timeliness, and relevance. Lastly, the third is the capacity of a CTI system to evaluate the overall quality of a CTI product and advertise it to its users.

Finally, the ***Collaboration*** layer refers to the capability of a CTI system to support the management of CTI operations. This capability handles the definition of requirements, coordination with other security groups, analysis control, and utilization of the Quality Control results. In this layer, we identify two functions: *CTI Operations Planning*, which is the capacity of a CTI system to support the planning of CTI operation from a team of analysts, and *Analysts Collaboration* which is the capacity of a CTI system to support instant collaboration among analysts that are internal or external to an organization.

5. Validation

As described in Section 3 (methodology), we follow the case study method to validate the proposed CTI reference model. More specifically, we use the actual cases to perform validation as close as possible to actual cybersecurity incidents:

1. the REvil gang attack on Quanta (Revil gang attack on Quanta);
2. the social engineering attack on Boshoku (Social engineering attack on Boshoku);
3. the DDoS attack launched against the Boston Children's Hospital (DDoS Case Study: DDoS Attack Mitigation Boston Children's Hospital).

Cyber security uses CTI in many circumstances. For example, CTI products are used as a source of data feeds in security defense mechanisms (e.g., IDS) or as a knowledge base during cyber incident investigation. Such diversity means that CTI systems may have different forms and complexity. Consequently, we chose the three case studies above to highlight the applicability of the proposed reference model in the design of various CTI systems. These case studies require the design of systems with different capabilities, while simultaneously related to different CTI scenarios or combinations of CTI scenarios (see Table 3).

Table 3. Case studies differences in relation to CTI scenarios.

	CTI Scenarios			
Case Study #	**1**	**2**	**3**	**4**
1	X		X	
2				X
3	X	X	X	X

5.1. Description of Case Studies

5.1.1. Case Study 1

The Chief Information Security Officer (CISO) of manufacturer A, a supplier of Apple Inc., has recently read a security report about the attack of the threat actor called REvil gang on Quanta, a manufacturer and primary supplier of Apple Inc. and other companies. To diminish the risk of a future incident, the CISO plans to use a CTI system that combines available public information of specific threat actors and automatically feeds the security systems of manufacturer A with its CTI products.

5.1.2. Case Study 2

After the social engineering attack incident at Toyota's subsidiary Boshoku Co., the management of Toyota's supplier A is willing to increase its personnel security awareness. To implement this decision, the company's CISO has decided to use a CTI system. This CTI system should collect CTI products related to social engineering attacks from public or private CTI sources and create reports and best practices comprehensible by the personnel of supplier A regardless of their technical background.

5.1.3. Case Study 3

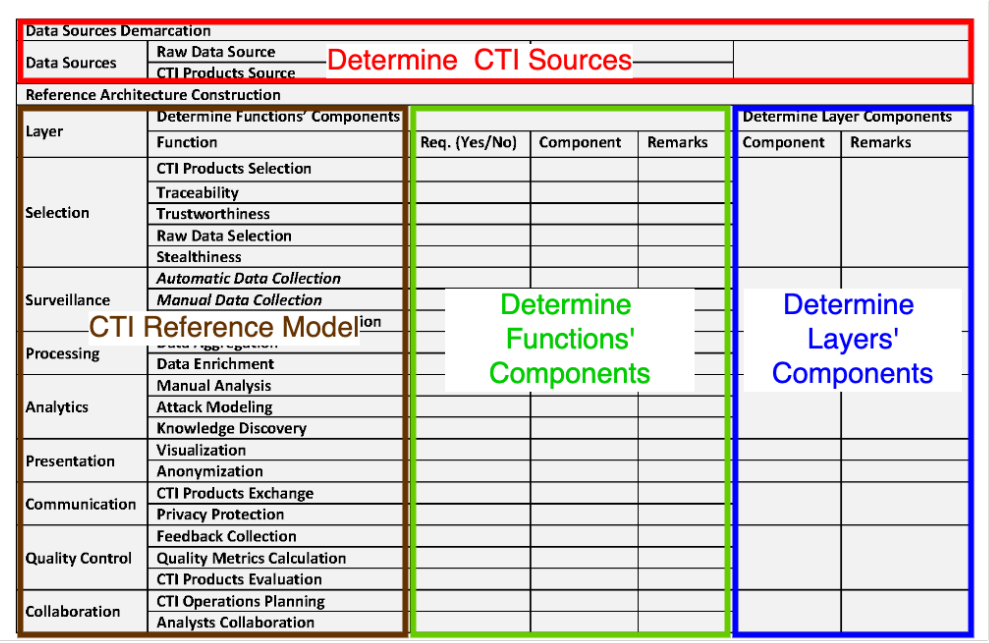
The cybersecurity team of Hospital Alpha, a healthcare institute belonging to a group of seven hospitals using the same internet service provider as Boston Children's Hospital, has been informed about a DDoS attack launched against the Boston Children's Hospital. To act proactively, the CISO of Hospital Alpha plans to use a CTI system to gather, analyze and use intelligence related to the Boston Children's Hospital incident. There are strong indicators that threat actors use the darknet for coordination, while the Boston Children's Hospital security team is positive about sharing information related to the incident. In addition, the CISO requires that their security systems directly use the CTI system's products. Lastly, the CISO needs the CTI system to provide reports for informing the C-suite of Hospital Alpha regarding the potential impact and cost of such an attack

5.2. Applying the CTI Reference Model to Case Studies

We employ the above case studies to validate the proposed CTI reference model. Specifically, we apply the CTI reference model to the requirements of the CTI Systems' case studies to design their reference architectures. By producing the CTI reference architectures, we demonstrate the applicability of the proposed CTI reference model to the design of a CTI system.

For this purpose, we developed a method that analyzes the systems requirements following a systematic, standardized approach that applies the proposed CTI reference model. Additionally, we created a matrix structure that summarizes our method and assists the CTI systems designer in developing the respective CTI reference architecture. The matrix structure consists of four parts, as depicted in Figure 11. In the first part (red rectangle), the designer determines the CTI sources from which the CTI system will collect data. In the second part (brown rectangle), the CTI reference model is outlined. In the third part (green rectangle), the designer selects Yes or No in the Requirements ("Req.") column for each closed-ended question on whether a requirement for a function exists (e.g., Is there any traceability requirement?). For each positive answer, the designer adds a component in the respective column (Component) and writes the requirement's details in the remarks column. Finally, in the fourth part (blue rectangle), the designer writes in the "Remarks" column those requirements that are not related to specific functions of a layer; and adds a particular component in the respective column for each such requirement. After identifying a CTI reference architecture's components, the designer has to decide how information flows between those components. The information flow, in general, follows the CTI reference model's layers, but the designer maintains complete flexibility.

Application examples of the method detailed above are Tables S1–S3, which help us analyze the requirements of the case studies, which help us analyze the requirements of each of the three case studies in this paper.



Data Sources Demarcation						
Data Sources	Raw Data Source					
	CTI Products Source					
Reference Architecture Construction						
Layer	Determine Functions' Components				Determine Layer Components	
	Function	Req. (Yes/No)	Component	Remarks	Component	Remarks
Selection	CTI Products Selection					
	Traceability					
	Trustworthiness					
	Raw Data Selection					
	Stealthiness					
Surveillance	*Automatic Data Collection*					
	Manual Data Collection					
	[illegible]ion					
Processing	Data Aggregation					
	Data Enrichment					
Analytics	Manual Analysis					
	Attack Modeling					
	Knowledge Discovery					
Presentation	Visualization					
	Anonymization					
Communication	CTI Products Exchange					
	Privacy Protection					
Quality Control	Feedback Collection					
	Quality Metrics Calculation					
	CTI Products Evaluation					
Collaboration	CTI Operations Planning					
	Analysts Collaboration					

Figure 11. Matrix Structure used for the application of the CTI Reference Model.

5.2.1. Application on Case Study 1

In case study 1, we can recognize the first and third CTI scenarios (cf. Section 4.3.2), by identifying the CTI source (e.g., public information) and the purpose of the threat intelligence process (e.g., feed the security systems). A CTI system for manufacturer A should address the requirements set by those scenarios in the specific case study. In Table S1, we apply the CTI reference model to identify and design the components and the data flows which results into the CTI reference architecture depicted in Figure 12.

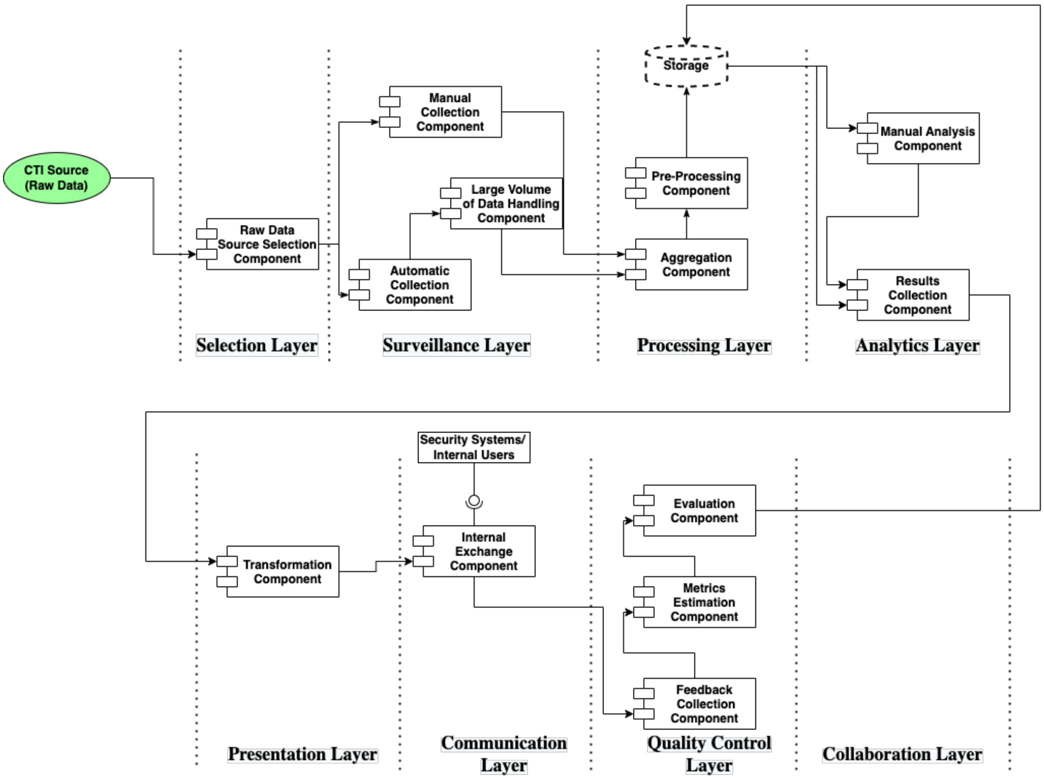

CTI Source (Raw Data)
Raw Data Source Selection Component
Manual Collection Component
Large Volume of Data Handling Component
Automatic Collection Component
Storage
Pre-Processing Component
Aggregation Component
Manual Analysis Component
Results Collection Component
Selection Layer
Surveillance Layer
Processing Layer
Analytics Layer
Security Systems/ Internal Users
Transformation Component
Internal Exchange Component
Evaluation Component
Metrics Estimation Component
Feedback Collection Component
Presentation Layer
Communication Layer
Quality Control Layer
Collaboration Layer

Figure 12. CTI Reference Architecture of Case Study 1.

5.2.2. Application on Case Study 2

In case study 2, we recognize the fourth CTI scenario. As in the previous case study, a CTI system for supplier A should address the requirements set by this scenario in the specific case study. In Table S2, we apply the CTI reference model to identify and design the components and the data flows which results into the respective CTI reference architecture, depicted in Figure 13.

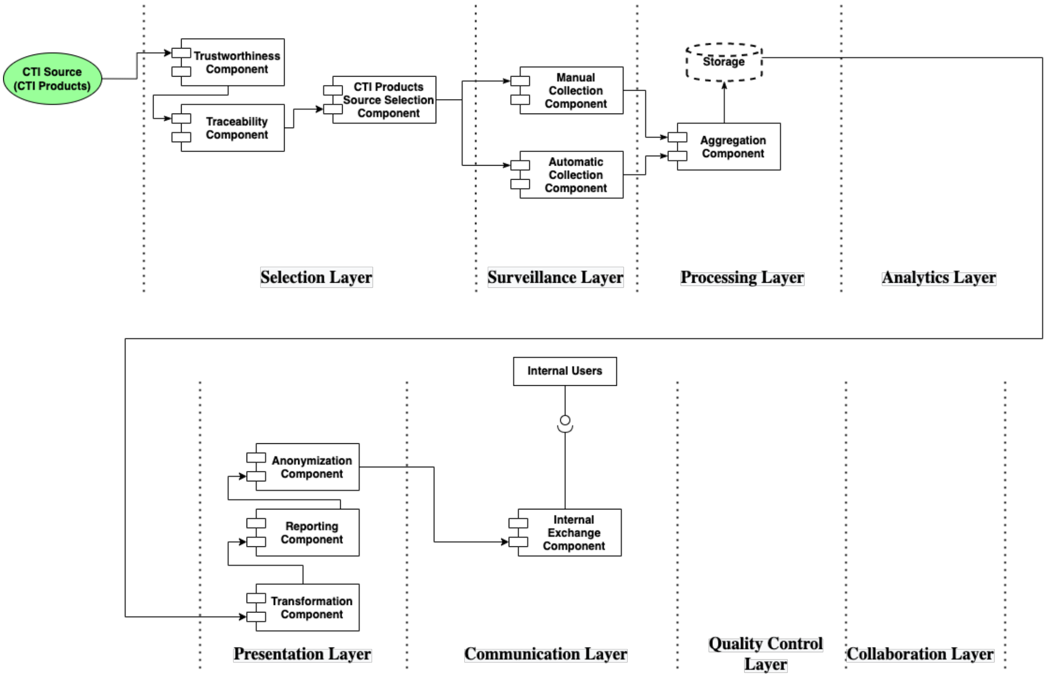


Figure 13. CTI Reference Architecture of Case Study 2.

5.2.3. Application on Case Study 3

A CTI system designed for Hospital Alpha must address the challenges set by case study 3, in which we recognize the four scenarios of Section 4.3.2. In Table S3, we apply the CTI reference model to identify and design the components and the data flows of the specific CTI system. The resulting CTI reference architecture is depicted in Figure 14.

5.3. Comparison of the Resulting CTI Architectures with Existing CTI Systems

Summarizing, for the use cases above, we validated the CTI reference model by demonstrating the design and development of the respective CTI reference architectures (Figures 12–14), for specific CTI systems. Furthermore, we illustrate the coverage of the case studies requirements of existing open-source CTI systems to reveal their architectural gaps demonstrating how the proposed CTI reference model contributes to a better CTI system design. In Table 4, we present the requirements coverage of specific CTI systems (in this research, we studied and analyzed twenty-three CTI systems; however, we present only the requirements coverage of open source CTI systems due to non-disclosure agreements).

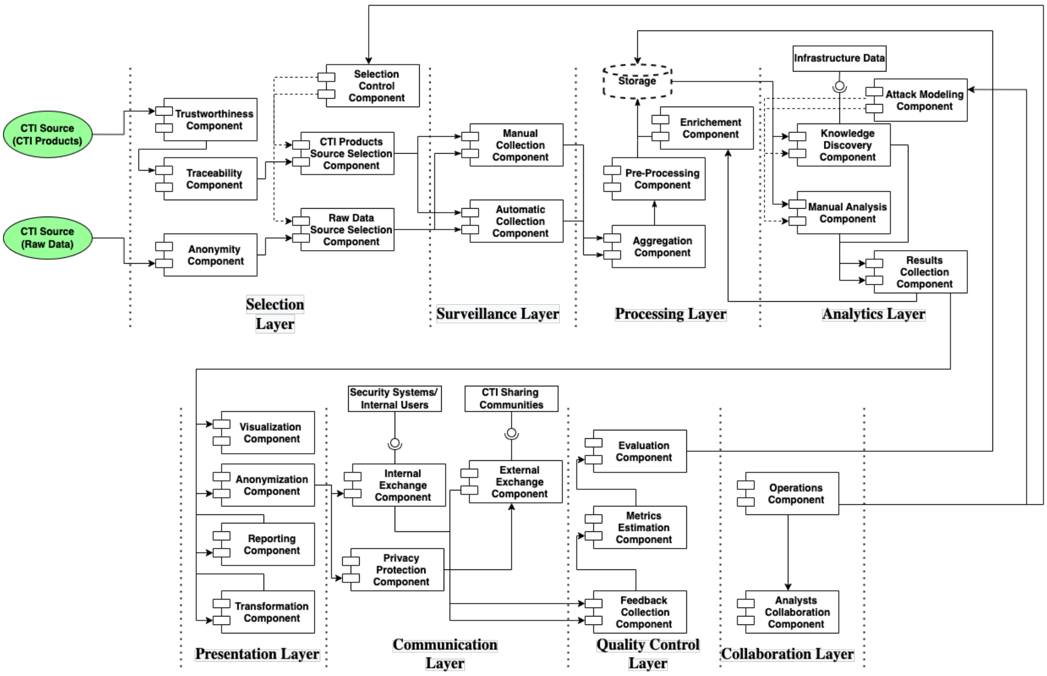

CTI Source (CTI Products)
CTI Source (Raw Data)
Trustworthiness Component
Traceability Component
Anonymity Component
Selection Control Component
CTI Products Source Selection Component
Raw Data Source Selection Component
Selection Layer
Manual Collection Component
Automatic Collection Component
Surveillance Layer
Storage
Enrichment Component
Pre-Processing Component
Aggregation Component
Processing Layer
Infrastructure Data
Attack Modeling Component
Knowledge Discovery Component
Manual Analysis Component
Results Collection Component
Analytics Layer
Visualization Component
Anonymization Component
Reporting Component
Transformation Component
Presentation Layer
Security Systems/ Internal Users
CTI Sharing Communities
Internal Exchange Component
External Exchange Component
Privacy Protection Component
Communication Layer
Evaluation Component
Metrics Estimation Component
Feedback Collection Component
Quality Control Layer
Operations Component
Analysts Collaboration Component
Collaboration Layer

Figure 14. CTI Reference Architecture of Case Study 3.

Table 4. Coverage of Requirements of CTI Case Studies by existing Open-source CTI Systems [81–83].

Layer	**Function**	**Case Studies**			**Open-Source CTI Systems**			**Requirement Coverage (%) by Open-Source CTI Systems**
		1	**2**	**3**	**YETI**	**MISP**	**CRITS**	
Selection	CTI Products Selection		X	X	X	X	X	100%
	Traceability		X	X	X	X	X	100%
	Trustworthiness		X	X	X	X	X	100%
	Raw Data Selection	X		X				0%
	Stealthiness			X				0%
Surveillance	Automatic Data Collection	X	X	X			X	33%
	Manual Data Collection	X	X	X	X	X		66%
	Large Volume of Data Collection	X						0%
Processing	Data Aggregation	X	X	X	X	X	X	100%
	Data Enrichment			X	X	X		66%
Analytics	Manual Analysis	X		X	X	X	X	100%
	Attack Modeling			X				0%
	Knowledge Discovery			X	X	X	X	100%
Presentation	Visualization			X	X	X		66%
	Anonymization			X				0%
Communication	CTI Products Exchange	X	X	X	X	X	X	100%
	Privacy Protection			X				0%
Quality Control	Feedback Collection			X		X		33%
	Quality Metrics Calculation			X				0%
	CTI Products Evaluation			X	X			33%
Collaboration	CTI Operations Planning	X		X				0%
	Analysts Collaboration	X		X				0%

6. Conclusions and Future Work

In this paper, we propose a CTI reference model following a reference model construction methodology. Our goal was to contribute towards the efficient analysis and design of CTI systems. For this reason, we analyzed the existing literature on CTI and CTI related standards.

To address the lack of clearly defined key concepts of CTI, we provided a set of corresponding definitions and developed a CTI frame of reference as an intermediate step in the CTI reference model construction process. Next, we combined the analysis results of twenty-three open-source and private CTI systems.

Furthermore, we systematically gathered and presented an analysis of complexity factors. To the best of our knowledge, there is no other work providing a thorough analysis of the complexity factors.

As part of the methodology's validation phase, we introduced a method for the use of the proposed CTI reference model in the design of CTI reference architectures with the following innovative characteristics:

- it introduces a systematic requirements analysis for the design of CTI systems' reference architectures;
- it integrates the CTI complexity factors in the CTI systems requirements analysis following a holistic approach to the design of CTI systems;
- it simplifies the way a CTI system's designer selects the components of the reference architecture by posing a set of closed-ended questions.

Next, we validated the CTI reference model following the case-study based approach.

Finally, our future work aims to apply the proposed CTI reference model, in order to design a CTI system focused on CTI from public sources—the darknet, in particular. We will also work towards developing CTI quality metrics and how they can be integrated into CTI systems.

Supplementary Materials: The following are available online at https://www.mdpi.com/article/10.3390/electronics11091401/s1, Table S1: Application of CTI Reference Model on Case Study 1, Table S2: Application of CTI Reference Model on Case Study 2, and Table S3: Application of CTI Reference Model on Case Study 3, Figure S1: CTI complexity factors concerning CTI frame of reference.

Author Contributions: Conceptualization, G.S. and P.F.; methodology, G.S.; validation, G.S., P.F. and I.M.; formal analysis, G.S.; investigation, G.S.; data curation, G.S.; writing—original draft preparation, G.S.; writing—review and editing, G.S, P.F., I.M. and P.S.; visualization, G.S., P.F. and I.M.; supervision, P.F., I.M. and P.S.; funding acquisition, P.S. All authors have read and agreed to the published version of the manuscript.

Funding: This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 833955.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bissell, K.; Fox, J.; LaSalle, R.M.; Cin, P.D. State of Cybersecurity Report 2021. Technical Report. Accenture Security. 2021. Available online: https://www.accenture.com/_acnmedia/PDF-165/Accenture-State-Of-Cybersecurity-2021.pdf (accessed on 5 April 2022).
2. Ardagna, C.; Corbiaux, S.; Sfakianakis, A.; Douligeris, C. *ENISA Threat Landscape 2021*; Technical Report; European Union Agency for Cybersecurity (ENISA): Athens, Greece, 2021.
3. X Force. IBM X-Force Threat Intelligence Index | IBM. Technical Report. IBM Security. 2020. Available online: https://www.ibm.com/downloads/cas/DEDOLR3W (accessed on 5 April 2022).
4. Accenture. *Third Annual State of Cyber Resilience Innovate for Cyber Resilience Lessons from Leaders to Master Cybersecurity Execution*; Technical Report; Accenture Security; Accenture: Dublin, Ireland, 2020.
5. Directorate-General for Communication; Leyen, U.v.d. A Union That Strives for More–Publications Office of the EU.2019. Available online: https://op.europa.eu/en/publication-detail/-/publication/43a17056-ebf1-11e9-9c4e-01aa75ed71a1 (accessed on 5 April 2022).
6. CheckPoint. Security Report 2020 | Check Point Software. 2020. Available online: https://resources.checkpoint.com/cyber-security-resources/cyber-security-report-2020 (accessed on 5 April 2022).
7. Ramsdale, A.; Shiaeles, S.; Kolokotronis, N. A Comparative Analysis of Cyber-Threat Intelligence Sources, Formats and Languages. *Electronics* **2020**, *9*, 824. [CrossRef]
8. Dauda, A.; Mclean, S.; Almehmadi, A.; El-Khatib, K. Big Data Analytics Architecture for Security Intelligence. In Proceedings of the 11th International Conference on Security of Information and Networks–SIN '18, Cardiff, UK, 10–12 September 2018; ACM Press: New York, NY, USA, 2018; pp. 1–4. [CrossRef]
9. Beard, C.; Brown, S.; Dulaunou, A.; Ginn, J.; Stipraro, P. *Exploring the Opportunities and Limitations of Current Threat Intelligence Platforms*; Technical Report; ENISA: Athens, Greece, 2017.
10. Tolstykh, T.; Gamidullaeva, L.; Shmeleva, N.; Lapygin, Y. Blockchain-Based Cyber Threat Intelligence System Architecture for Sustainable Computing. *Sustainability* **2020**, *12*, 6401. [CrossRef]
11. DTIC. *JP 2-0 Joint Intelligence*; US Department of Defense: Fort Lee, VA, USA, 2007; pp. 1–144.

12. de Melo e Silva, A.; Gondim, J.J.C.; de Oliveira Albuquerque, R.; Villalba, L.J.G. A Methodology to Evaluate Standards and Platforms within Cyber Threat Intelligence. *Future Internet* **2020**, *12*, 108. [CrossRef]
13. Bauer, S.; Fischer, D.; Sauerwein, C.; Latzel, S.; Stelzer, D.; Breu, R. Towards an evaluation framework for threat intelligence sharing platforms. In Proceedings of the Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 7–10 January 2020; pp. 1947–1956. [CrossRef]
14. Camarinha-Matos, L.M.; Afsarmanesh, H. Reference modeling: Needs and basic terminology. In *Collaborative Networks: Reference Modeling*; Springer: Boston, MA, USA, 2008; Chapter 2, pp. 33–50. [CrossRef]
15. Thomas, O. Version management for reference models: Design and implementation. In *Reference Modeling: Efficient Information Systems Design Through Reuse of Information Models*; Physica-Verlag HD: Heidelberg, Germany, 2007; pp. 1–26. [CrossRef]
16. Schmid, B.; Lindemann, M. Elements of a Reference Model for Electronic Markets. In *Thirty-First Annual Hawaii International Conference on System Sciences-Volume 4*; IEEE Computer Society: St. Gallen, Switzerland, 1998; pp. 193–201.
17. Helm, J. RUP Artifact: Reference Architecture. 2001. Available online: https://sceweb.uhcl.edu/helm/RationalUnifiedProcess/process/artifact/ar_refarch.htm (accessed on 5 April 2022).
18. Rosemann, M.; van der Aalst, W.M.P. A configurable reference modelling language. *Infor. Syst.* **2007**, *32*, 1–23. [CrossRef]
19. Shackleford, D. *CTI in Security Operations: SANS 2018 Cyber Threat Intelligence Survey*; SANS Institute: Bethesda, MD, USA, 2018.
20. EC-Counsil. *Certified Threat Intelligence Analyst*; EC-Council: Albuquerque, NM, USA, 2018.
21. Ahlemann, F.; Gastl, H. Process Model for an Empiracally Grounded Reference Model Construction. In *Reference Modeling for Business Systems Analysis*; Fettke, P., Loos, P., Eds.; IGI Global: Hershey, PA, USA, 2006; pp. 77–97. [CrossRef]
22. Pajk, D.; Indihar-Stemberger, M.; Kovacic, A. Reference model design: An approach and its application. In Proceedings of the ITI 2012 34th International Conference on Information Technology Interfaces, Cavtat, Croatia, 25–28 June 2012; IEEE: New York, NY, USA, 2012; pp. 455–460. [CrossRef]
23. IBM. UML Model Elements. Available online: https://www.ibm.com/docs/en/rational-soft-arch/9.7.0?topic=models-uml-model-elements (accessed on 1 April 2022).
24. Schuette, R.; Rotthowe, T. The guidelines of modeling—An approach to enhance the quality in information models. In *Conceptual Modeling—ER'98*; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1507, pp. 240–254. [CrossRef]
25. Merriam-Webster. Frame of Reference. Available online: https://www.merriam-webster.com/dictionary/frame (accessed on 1 April 2022).
26. Fettke, P.; Loos, P. Multiperspective evaluation of reference models—Towards a framework. In *International Conference on Conceptual Modeling*; Springer: Berlin/Heidelberg, Germany, 2003. [CrossRef]
27. Dalziel, H. *How to Define and Build an Effective Cyber Threat Intelligence Capability*; Syngress, an Imprint of Elsevier: London, UK, 2015.
28. Chismon, D.; Ruks, M. *Threat Intelligence: Collecting, Analysing, Evaluating*; Technical Report; MWR InfoSecurity: London, UK, 2015.
29. Tounsi, W.; Rais, H. A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Comput. Secur.* **2018**, *72*, 212–233. [CrossRef]
30. Ahrend, J.M.; Jirotka, M.; Jones, K. On the collaborative practices of cyber threat intelligence analysts to develop and utilize tacit Threat and Defence Knowledge. In Proceedings of the 2016 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (CyberSA),London, UK, 13–14 June 2016; IEEE: New York, NY, USA, 2016; pp. 1–10. [CrossRef]
31. Gundert, L. *Producing a World-Class Threat Intelligence Capability*; Technical Report; Recorded Future: Somerville, MA, USA, 2016.
32. Ernst & Young Global Limited. Cyber Threat Intelligence—How To Get Ahead Of Cybercrime. In *Insights on Goverance, Risk and 686 Compliance*; Ernst and Young: London, UK, 2014; Volume 1, pp. 1–16.
33. Jasper, S.E. Cyber Threat Intelligence Sharing Frameworks. *Int. J. Intell. Count.* **2017**, *30*, 53–65. [CrossRef]
34. Caltagirone, S.; Pendergast, A.; Betz, C. *The Diamond Model of Intrusion Analysis*; Technical Report; Defense Technical Information Center: Fort Belvoir, VA, USA, 2013.
35. Lockheed Martin. Cyber Kill Chain®|Lockheed Martin. Available online: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyberkill-chain.html (accessed on 24 March 2022)
36. Can, M. *Joint Intelligence Doctrine*; Canadian Forces: Ottawa, ON, Canada, 2003; p. 100.
37. Mod, U.K. *Understanding and Intelligence Support to Joint Operations (JDP 2-00)*; Joint Doctrine Publication: Bicester, UK, 2011; Volume 3, p. 155.
38. Davies, P.; Gustafson, K.; Ridgen, I. The Intelligence Cycle is dead, long live the Intelligence Cycle: Rethinking intelligence fundamentals for a new intelligence doctrine. In *Understanding the Intelligence Cycle*; Phythian, M., Ed.; Routledge: Leicester, UK, 2013; pp. 67–105.
39. Gill, P.; Phythian, M. From Intelligence Cycle to web of intelligence. In *Understanding the Intelligence Cycle*; Phythian, M., Ed.; Routledge: Leicester, UK, 2015; pp. 35–54.
40. Aviad, A.; Węcel, K. Cyber Treat Intelligence Modeling. In *Business Information Systems*; Abramowicz, W., Corchuelo, R., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 361–370. [CrossRef]
41. Skopik, F.; Settanni, G.; Fiedler, R. A problem shared is a problem halved: A survey on the dimensions of collective cyber defense through security information sharing. *Comput. Secur.* **2016**, *60*, 154–176. [CrossRef]
42. Sullivan, C.; Burger, E. In the public interest: The privacy implications of international business-to-business sharing of cyber-threat intelligence. *Comput. Law Secur. Rev.* **2017**, *33*, 14–29. [CrossRef]

43. Sillaber, C.; Sauerwein, C.; Mussmann, A.; 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'16, Vienna, Austria, 24 October 2016; ACM Press: New York, NY, USA, 2016; pp. 65–70. [CrossRef]
44. Wagner, T.D.; Mahbub, K.; Palomar, E.; Abdallah, A.E. Cyber threat intelligence sharing: Survey and research directions. *Comput. Secur.* **2019**, *87*, 101589. [CrossRef]
45. Menges, F.; Sperl, C.; Pernul, G. Unifying Cyber Threat Intelligence. In *Trust, Privacy and Security in Digital Business*; Gritzalis, S., Weippl, E., Katsikas, S., Anderst-Kotsis, G., Tjoa, M., Khalil, I., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 161–175. [CrossRef]
46. Mavroeidis, V.; Bromander, S. Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. In Proceedings of the 2017 European Intelligence and Security Informatics Conference (EISIC), Athens, Greece, 11–13 September 2017; IEEE: Athens, Greece, 2017; pp. 91–98. [CrossRef]
47. Skopik, F.; Qin, L. Trustworthy incident information sharing in social cyber defense alliances. In Proceedings of the 2013 IEEE Symposium on Computers and Communications (ISCC), Split, Croatia, 7–10 July 2013; IEEE: Split, Croatia, 2013; pp. 233–239. [CrossRef]
48. Peterson, J.J. *Appropriate Factorsto Consider When Assessing Analytics Confidence in Intelligence Analysis*; Technical Report; Mercyhurst College Institute for Intelligence Studies (MCIIS): Erie, PA, USA, 2008.
49. Obitade, P.O. Big data analytics: A link between knowledge management capabilities and superior cyber protection. *J. Big Data* **2019**, *6*, 71. [CrossRef]
50. Al-Mohannadi, H.; Mirza, Q.; Namanya, A.; Awan, I.; Cullen, A.; Disso, J. Cyber-Attack Modeling Analysis Techniques: An Overview. In Proceedings of the 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), Vienna, Austria, 22–24 August 2016; IEEE: Vienna, Austria, 2016; pp. 69–76. [CrossRef]
51. Ullah, F.; Babar, M.A. Architectural Tactics for Big Data Cybersecurity Analytics Systems: A Review. *J. Syst. Softw.* **2019**, *151*, 81–118. [CrossRef]
52. Pacheco, F.; Exposito, E.; Gineste, M.; Baudoin, C.; Aguilar, J. Towards the deployment of Machine Learning solutions in network traffic classification: A systematic survey. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 1988–2014. [CrossRef]
53. Zuech, R.; Khoshgoftaar, T.M.; Wald, R. Intrusion detection and Big Heterogeneous Data: A Survey. *J. Big Data* **2015**, *2*, 41. [CrossRef]
54. Lankau, J.; Smith, K.; Deason, L.; Geide, M.; Baxter, J. *Lessons Learned From Data Science Application to Cyber Security Network Logs*; Technical Report; Punch Cyber Analytics Group: Reston, VA, USA, 2018.
55. Settanni, G.; Shovgenya, Y.; Skopik, F.; Graf, R.; Wurzenberger, M.; Fiedler, R. Acquiring cyber threat intelligence through security information correlation. In Proceedings of the 2017 3rd IEEE International Conference on Cybernetics (CYBCONF), Exeter, UK, 21–23 June 2017; Institute of Electrical and Electronics Engineers Inc.: Vienna, Austria, 2017; pp. 1–7. [CrossRef]
56. Iqbal, Z.; Anwar, Z.; Mumtaz, R. STIXGEN—A Novel Framework for Automatic Generation of Structured Cyber Threat Information. In Proceedings of the 2018 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 17–19 December 2018; IEEE: Islamabad, Pakistan, 2018; pp. 241–246. [CrossRef]
57. Zhang, H.; Yi, Y.; Wang, J.; Cao, N.; Duan, Q. Network security situation awareness framework based on threat intelligence. *Comput. Mater. Contin.* **2018**, *56*, 381–399. [CrossRef]
58. Menges, F.; Pernul, G. A comparative analysis of incident reporting formats. *Comput. Secur.* **2018**, *73*, 87–101. [CrossRef]
59. Casey, E.; Barnum, S.; Griffith, R.; Snyder, J.; van Beek, H.; Nelson, A. The Evolution of Expressing and Exchanging Cyber-Investigation Information in a Standardized Form. In *Handling and Exchanging Electronic Evidence Across Europe*; Springer: Cham, Switzerland, 2018; pp. 43–58. [CrossRef]
60. Cheng, Y.; Deng, J.; Li, J.; DeLoach, S.A.; Singhal, A.; Ou, X. Metrics of Security. In *Cyber Defense and Situational Awareness*; Kott, A.,Wang, C., Erbacher, R., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 263–295. [CrossRef]
61. Grispos, G.; Glisson, W.B.; Storer, T. How Good is Your Data? Investigating the Quality of Data Generated During Security Incident Response Investigations. In Proceedings of the 52nd Hawaii International Conference on System Sciences, Maui, HI, USA, 4–7 January 2012; Scholar Space Hawaii International: Honolulu, HI, USA, 2019; p. 10. [CrossRef]
62. Friedman, J.A.; Zeckhauser, R. Assessing uncertainty in intelligence. *Int. J. Inf. Secur.* **2012**, *27*, 824–847. [CrossRef]
63. Schlette, D.; Böhm, F.; Caselli, M.; Pernul, G. Measuring and visualizing cyber threat intelligence quality. *Int. J. Inf. Secur.* **2020**, *20*, 1–18. [CrossRef]
64. Virustotal. YARA—The Pattern Matching Swiss Knife for Malware Researchers. Available online: https://virustotal.github.io/yara/ (accessed on 20 Febuary 2022)
65. MITRE. CWE–Common Weakness Enumeration. Available online: https://cwe.mitre.org/ (accessed on 2 February 2022).
66. MITRE. CVE–CVE. Available online: https://cwe.mitre.org/ (accessed on 20 February 2022).
67. NIST. NVD–CCE. Available online: https://nvd.nist.gov/config/cce (accessed on 20 February 2022).
68. NIST. NVD– CPE. Available online: https://nvd.nist.gov/products/cpe (accessed on 20 February 2022).
69. MITRE. About MAEC | MAEC Project Documentation. Available online: https://maecproject.github.io/about-maec/ (accessed on 20 February 2022).
70. MITRE. CAPEC–Common Attack Pattern Enumeration and Classification (CAPEC™). Available online: https://capec.mitre.org/ (accessed on 20 February 2022).

71. MITRE. MITRE ATT&CK®. Available online: https://attack.mitre.org/ (accessed on 20 February 2022).
72. MITRE. CybOX–Cyber Observable Expression | CybOX Project Documentation. Available online: https://cyboxproject.github.io/ (accessed on 20 February 2022).
73. OASIS. Introduction to STIX. Available online: https://oasis-open.github.io/cti-documentation/stix/intro (accessed on 20 February 2022).
74. Gibb, W.; Kerr, D. OpenIOC: Back to the Basics | FireEye Inc. Available online: https://www.fireeye.com/blog/threat-research/2013/10/openiocbasics.html (accessed on 20 February 2022).
75. FIRST. Traffic Light Protocol (TLP). Available online: https://www.first.org/tlp/ (accessed on 20 February 2022).
76. OASIS. Introduction to TAXII. Available online: https://oasis-open.github.io/cti-documentation/taxii/intro (accessed on 20 February 2022).
77. IETF. RFC 7970—The Incident Object Description Exchange Format Version 2. Available online: https://datatracker.ietf.org/doc/rfc7970/ (accessed on 20 February 2022).
78. VerisCommunity. The VERIS Framework. Available online: http://veriscommunity.net/ (accessed on 20 February 2022).
79. Bass, L.; Clements, P.; Kazman, R. *Software Architecture in Practice*, 3rd ed.; Addison-Wesley Professional: London, UK, 2013.
80. The Open Group. *The Open Group Architecture Framework (TOGAF) Version 9*; The Open Group: San Francisco, CA, USA, 2009.
81. YETI. Available online: https://yeti-platform.github.io/ (accessed on 20 February 2022).
82. MISP. MISP Open Source Threat Intelligence Platform and Open Standards For Threat Information Sharing. Available online: https://www.mispproject.org/ (accessed on 20 February 2022).
83. CRITS. CRITs: Collaborative Research Into Threats. Available online: https://crits.github.io/ (accessed on 20 February 2022).