Investigating Sharing of Cyber Threat Intelligence and Proposing A New Data Model for Enabling Automation in Knowledge Representation and Exchange

SIRI BROMANDER, mnemonic/University of Oslo

MORTON SWIMMER, Trend Micro Research

LILLY PIJNENBURG MULLER, King's College

AUDUN JØSANG, University of Oslo

MARTIN EIAN, GEIR SKJØTSKIFT, and FREDRIK BORG, mnemonic

For a strong, collective defense in the digital domain, we need to produce, consume, analyze, and share cyber threat intelligence. With an increasing amount of available information, we need automation to ensure adequate efficiency. We present the results from a questionnaire investigating the use of standards and standardization and how practitioners share and use cyber threat intelligence (CTI). We propose a strict data model for CTI that enables consumption of all relevant data, data validation, and analysis of consumed content. The main contribution of this article is insight into how CTI is shared and used by practitioners, and the strictness of the data model that enforces input of information and enables automation and deduction of new knowledge.

CCS Concepts: • **Security and privacy**; • **Theory of computation** → Semantics and reasoning; • **Information systems** → *Expert systems*; Data analytics; *Graph-based database models*; • **Applied computing** → **Evidence collection, storage and analysis**; *Investigation techniques*;

Additional Key Words and Phrases: Cyber threat intelligence, security, knowledge graph, ontology

ACM Reference format:

Siri Bromander, Morton Swimmer, Lilly Pijnenburg Muller, Audun Jøsang, Martin Eian, Geir Skjøtskift, and Fredrik Borg. 2021. Investigating Sharing of Cyber Threat Intelligence and Proposing A New Data Model for Enabling Automation in Knowledge Representation and Exchange. *Digit. Threat. Res. Pract.* 3, 1, Article 6 (October 2021), 22 pages.
<https://doi.org/10.1145/3458027>

1 INTRODUCTION

In recent years, we have seen several initiatives to structure and streamline **Cyber Threat Intelligence (CTI)**. Organizations share CTI in a variety of ways, and most of the CTI in an average organization comes from

Authors' addresses: S. Bromander, mnemonic/University of Oslo, Oslo, Norway; email: siri@mnemonic.no; M. Swimmer, Trend Micro Research, Munich, Germany; email: morton_swimmer@trendmicro.com; L. Pijnenburg Muller, King's College, London, United Kingdom; email: lilly.muller@kcl.ac.uk; A. Jøsang, University of Oslo, Norway; email: audun.josang@mn.uio.no; M. Eian, G. Skjøtskift, and F. Borg, mnemonic, Oslo, Norway; emails: {meian, geir, fredrikb}@mnemonic.no.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.

2576-5337/2021/10-ART6 \$15.00

<https://doi.org/10.1145/3458027>

external sources. Consuming, normalizing, enriching, and analyzing CTI from heterogeneous sources are major challenges for CTI analysts.

Gartner defines *threat intelligence* as “evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice, about an existing or emerging menace or hazard to assets that can be used to inform decisions regarding the subject’s response to that menace or hazard” [15]. The key phrase in this definition is “evidence-based knowledge,” as CTI needs to be based upon evidence in order to be trusted. It is also of importance that this type of knowledge can be used for mitigating threats and hence needs to be actionable. The Allied Joint Procedures published and used by NATO [1] concur with this definition of threat intelligence, elaborating on the differences between threat data, threat information, and threat intelligence. Threat data can be elevated to become threat information, which needs structure and adoption for a given audience to qualify as threat intelligence.

Available standards for structuring and automating CTI are available, but the use thereof seems limited. Successful defense against threats depends on effective use of a scarce resource: our security experts. This shortage of personnel in cyber security means we need to take advantage of any available technical aid, increasing the automation of repetitive tasks that require little cognitive analysis or human evaluation. We also find that automation makes the available CTI more useful for big-data analytics and advanced reasoning, leading to significantly strengthened analysis capabilities, insight, and decision support. However, this relies on having consistent and well-structured data.

This article questions whether the current state of CTI adequately enables practitioners to utilize automation. We have investigated how practitioners execute and understand CTI, and propose the **Semi-Automated Cyber Threat Intelligence (ACT)** data model to address the automation challenges.

1.1 Research Questions and Motivation

To stimulate development in the field of CTI, we need a structured way of representing data, information, and knowledge about cyber threats. Several initiatives exist in this regard, and the one gaining the most attention has been STIX (Structured Threat Information Expression). STIX now consists of 18 different **STIX Domain Objects (SDOs)** and two different relationship types that have a list of suggestions for relationship names and SDOs they may connect. STIX is argued to be the de facto standard for representing CTI within the technical CTI community [30]. Yet although STIX is the most used standard for representing CTI, this article questions whether the features STIX is praised for are actually used in practice.

Preliminary research indicates that the amount of shared CTI that is not based on standards is indeed high, with possible impact on the development of new standards, models, software, and processes of CTI. Sauerwein et al. [29] confirm this. From this initial analysis, the following research questions emerged. How do practitioners understand CTI, and how do they share it? Do their understanding and sharing of CTI align with the current development of the field of tools and formal representations of CTI?

Furthermore, if our suspicion about low use of standards for CTI is correct, how can we contribute to achieving the goal of automating the CTI process? We saw the need for a data model that enables automation and analysis of our available threat intelligence and satisfies the following criteria to provide practical usability:

- *Enrichment and analysis*: Combining all available data from different data sources will increase the analysis capability of an analyst and remove repetitive tasks.
- *Expressiveness*: The import and export of CTI from a system can be trivial if the data is stored in a consistent and structured manner, covering all relevant data. How we model our data storage is hence critical.
- *Format consistency*: A key requirement for automation and analysis is data quality, which includes both content and format. We cannot enforce quality of content, but we can enable an analyst to evaluate this. Format consistency can be enforced by a strict data model. This means that a computer knows where to

Fig. 1. The pyramid of pain [4].

Fig. 2. A slightly expanded DML model [7].

find a certain data type in a dataset, and that the data found in that place always is the same type of data. If we allow (too much) flexibility within the schema of a data model, this requirement will not be met, removing the ability to automate consumption and analysis across different platforms.

- *Schema and restrictions to express knowledge:* Threat intelligence analysis traditionally requires a large amount of knowledge from the analyst. Adding knowledge into the data model will make the knowledge available to more analysts. This requires that the data model is based on schema and type restrictions.

Threat intelligence depends on collaboration between a range of organizations and communities. Any tool or system used by collaborators should be openly available to the community without restrictions, which has been a key motivation for this project.

2 BACKGROUND AND RELATED WORK

Several publications describe aspects of how people conduct/perform CTI-related tasks. Sauerwein et al. [30] indicate that STIX is the de facto standard for sharing CTI, and in another work [29] they further describe sharing to be mostly executed in unstructured manners. Ramsdale et al. [28] conclude that there is no agreed and standardized way of sharing CTI, much in line with the findings presented in this article.

2.1 Models Describing CTI

In “The Pyramid of Pain,” Bianco [4] presents a model for representation of CTI-relevant data, as seen in Figure 1. The model is often cited where CTI is discussed. The pyramid emphasizes where countermeasures can be applied so as to impede (i.e., inflict pain) the adversary in the most effective manner. At the bottom of the pyramid are hash values of malware samples. Although these are simple for defenders to detect and block, they are equally simple for the attackers to mutate. Conversely, at the top of the pyramid, Bianco places **Tactics, Techniques, and Procedures (TTP)** as the hardest for the attackers to change but also the hardest for defenders to identify. TTP is a concept with a long tradition in intelligence, but which is challenging to use within CTI, as it is difficult to identify what the exact tactic, technique, and/or procedure is when sharing and (automatically) processing CTI. The development of capabilities for detecting threats based on TTPs is nevertheless of utmost importance for further development of our collective defense in the cyber domain.

In 2014, Stillions [32] published a similar model, the **Detection Maturity Level (DML)** model seen in Figure 2. Stillions’ DML model specifies hierarchical levels/types of data an organization should be able to consume to be at a given level. In this case, to be “able to consume” means to not only receive and understand the data or information but also to be able to take useful action based on it. The model has proven to be useful in several contexts, such as for describing the relevant data exchanged within CTI [7], where tactics, techniques, and procedures are behavioral indicators.

Fig. 3. CTI as described by Chismon and Ruks [10].

The model of Chismon and Ruks [10], illustrated in Figure 3, represents the current types of CTI and to what degree they are detailed and of long-term use. This model illustrates that the more detailed the knowledge is, the more certainty about the threat actor's presence and identity can be obtained, and the more robust (long term) the model is, the longer such knowledge is useful for the defenders. Tactical threat intelligence consisting of TTPs is of most value in the attempt to detect and prevent future attacks. Yet this is in our experience one of the least developed areas within CTI and also the area currently receiving the most attention in research and development within CTI.

2.2 Structuring CTI

There have been several attempts at structuring CTI. The motivations for the different approaches seem to differ, which also influences the results. Structuring CTI often results in an ontology, even though it may not be referred to as one. The following standards, formats, and platforms are all containing their own ontology. A description of ontology and example of published ontologies are separated in the next section for increased readability.

Barnum [3] proposed the STIX format in 2012. The intention was to enable the *sharing* of CTI in a more structured format than plain text. STIX was intended as a data exchange format and not a data storage model. STIX was published in version 2.0 in 2017 [24] and is now in version 2.1.

The **Malware Information Sharing Platform (MISP)** [36] is a platform for rapid sharing of indicators of compromise and sightings of indicators. The MISP data model is under continuous development [22]. The data contained within MISP platforms correspond well with the suggested data model in this article; however, the popularity of the platform and loose data governance has led, in our experience, to a decrease in the consistency of the data.

ATT&CK [23] is a framework and knowledge base for describing adversary behavior through enumerating adversary groups, tactics, techniques, and tools and the relationships between them. The knowledge base is maintained by MITRE, and it is published online. ATT&CK uses a data model with defined relationships for structuring their knowledge base.

In 2017, the ACT project¹ was initiated and is now a platform allowing for consumption, analysis, enrichment, and sharing of CTI. The research presented in this article has been part of the development of the ACT platform as described in the work of Bromander et al. [8].

In the spring of 2019, OpenCTI [2] was published. The platform has similar ideas as the ACT project, building a graph-based platform with strong query possibilities, where a combination of sources is handled and where the data model enables linked data. The chosen open source license² deviates from the ACT platform.³

2.3 Ontologies

An ontology, in the field of computer science, is a formal description of concepts and how they are related to each other, often referred to as classes and properties. In turn, ontologies provide computational meaning to data by building semantic and logic relations in the ontology that enables us to use reasoning methods (e.g., induction or deduction) on our data in our knowledge base. Although there are many implementations of knowledge bases and ontologies, the World Wide Web Consortium (W3C) chose a triplet model for facts and calls this the **Resource Description Framework (RDF)**.⁴ RDF also allows us to implement the RDFS schema language⁵ and OWL,⁶ the ontology language.

Triples take the form of subject, predicate, and object and are expressed as URIs.⁷ For example, in English we can say that *FQDN www.example.com resolves to A record 192.168.1.2*. In RDF, this example could look like `<http://icann.org/FQDN#www.example.com> <http://ietf.org/dns#resolvesToA> <http://icann.org/ipv4#192.168.1.2>`. Note that these URIs only need to be unique and defined in some knowledge base. Typically, we use XML namespaces⁸ to simplify this to `fqdn:www.example.com dns:RESOLVESToA ipv4:192.168.1.2`. Continuing the example, we can say that `www.example.com` is an instance of a fully qualified domain name by stating `fqdn:www.example.com rdf:type dns:FQDN`. Together with axioms and rules, we can create a knowledge base with reasoning capabilities.

There are several ontologies built with the aim to structure security relevant data. They cover a range of data and motivations like data validation, transformation, or logical reasoning. An overview of available ontologies may be found in the work of Mavroeidis and Bromander [19]. To the extent of our knowledge, none of the available ontologies are alone suitable for solving our problem; however, the UTIM⁹ ontology is being developed in parallel with the model presented in this article, and it is hoped that one day data from the ACT model will be freely interchangeable with data modeled with UTIM. Menges et al. [21] propose a unified CTI model that covers the concepts used within CTI. This work should be tested with data and in practical use. The lack of defined differences in relationships in this model may be a challenge when expressing knowledge and makes it closer to a taxonomy than an ontology.

The rest of the article is structured as follows. First we describe the methodology of our work in Section 3. Then we present the questionnaire results and discuss them in Section 4. The details of the data model are explained in Section 5, which includes a graph representation of the data model. We discuss our findings in Section 5.5 and conclude in Section 6.

¹<https://github.com/mnemonic-no/act-platform>.

²<https://github.com/OpenCTI-Platform/opencti/blob/master/LICENSE>.

³<https://github.com/mnemonic-no/act-platform/blob/master/LICENSE>.

⁴<https://www.w3.org/TR/rdf-concepts/>.

⁵<https://www.w3.org/TR/rdf-schema/>.

⁶<https://www.w3.org/TR/owl2-overview/>.

⁷<https://tools.ietf.org/html/rfc2396>.

⁸<https://www.w3.org/TR/xml-names/>.

⁹<http://www.ti-semantics.com>.

3 METHODOLOGY

This research work is based on triangulation of a preliminary ethnographic observation of CTI environments, a questionnaire, and semi-structured interviews. In addition, Eclectic IQ performed and presented a statistical analysis on STIX usage in the work of Polzunov and Abraham [27]. These results are used in the discussion together with our own results. The most relevant results from their study are described in Sections 4.1.2 and 4.2.1.

For our questionnaire, we have used the work of Krosnick [18] and Smyth et al. [31] as guidance, with the most important choices listed as follows:

- To minimize the respondents' fatigue, we chose to keep the number of questions as low as possible. This limits the amount of information we can extract but presumably increases the quality of the results.
- We chose questions that were easy to reply to, requiring a limited amount of interpretation of the questions, and also a limited amount of retrieval of relevant information in their memory or integration of this information in judgments and final answers (choice of option).
- When we asked the respondents to estimate percentages, we chose to make the questions open to escape biases. We chose the classes to be reported based upon the answers we got.
- We chose open questions where possible to mitigate possible biased responses in the case of a non-exhaustive list. The use of "Other, please specify" is not recommended as a solution to this according to Krosnick [18] and hence not used.
- Experimental evidence suggests that checklists should be structured in "did"/"did not" format as opposed to "check-all-that-apply" partially because respondents take longer to answer forced choice items and then are more likely to think before answering, and partially because forced-choice answers are easier to interpret [31]. This is the argument for choosing this option in our questionnaire.

The questionnaire was developed in three stages, with an initial version tested on a reference group of four people. Improvements and a new version was tested on a new reference group of 11 persons during the 2019 FIRST CTI Symposium. The final version was created with the input and evaluation of this second version. To set the frame for the questionnaire, a limited text was included to introduce it. A text version of the questionnaire can be found in the work of Bromander [6]. The questionnaire was published using the web portal Nettskjema provided by the University of Oslo [26]. The questionnaire was online from June 15 to August 8, 2019. There is a limited number of CTI professionals, so the method for attracting many as possible to participate was to advertise in public and to time the data collection during the 31st Annual FIRST Conference in June 2019 where many CTI practitioners met. Participants were volunteers, which may influence the selection. The distribution of participants by country is shown in Figure 4.

Interviews were conducted post-questionnaire, with three participants. The participants were volunteers since the respondents were anonymous and hence could not be asked or selected for participation. The participants were asked to comment on the questionnaire questions, and the answers were used to better understand the results. The interview guide can be found in the work of Bromander [6]. As part of the interviews, the participants were asked to represent a given piece of information in STIX, used as a case study in this article.

The data model has been developed using an iterative process. The development of the data model has been data driven, meaning that we have developed, updated, and tested the model based on relevant available data.

3.1 Limitations

Although the data model is an ontology, it is not implemented in RDFS or OWL, but all content can be exported as triplets. Initial testing of implementing the data model using Protégé¹⁰ has been done to find improvements,

¹⁰<https://protege.stanford.edu>.

Fig. 4. Respondents by country.

but the desired reasoning capabilities lead to the need for rule-based reasoning, which can be performed on top of the proposed data model with other tools as well.

To avoid inconsistent data in the knowledge base, a strict data model is needed. The proposed data model requires a certain amount of work to consume new data sources because of this chosen strictness.

4 QUESTIONNAIRE RESULTS AND DISCUSSION

The following section is presenting the results from the questionnaire and follow-up interviews and discussions following the results.

The semi-structured interviews contributed to the choice of included results in Section 4.2.1, and to the discussions found in Sections 4.2.2 through 4.2.5.

Section 4.2.1 includes the results of a case study to explain the problem with flexibility within STIX, performed as part of the semi-structured interviews.

The definition of sectors and organization sizes are found elsewhere [11, 16]. There were 36 respondents to the complete questionnaire. The relatively small number of respondents means that all results must be seen as indicative, not as conclusive. All percentages are rounded off, and thus not all of the pie charts sum to 100%.

Of the 36 respondents, 30 answered that they shared CTI as part of their role. The respondents were representing the sectors, organization size, and countries as seen in Figures 4, 5, and 6.

4.1 Questionnaire Results

4.1.1 Sharing CTI. The main indications possible to draw from the responses are related to the format of shared CTI, the use of consumed CTI, and the reasons for not sharing. All the respondents considered themselves consumers of CTI, and approximately 70% considered themselves producers of CTI:

- *Format of shared CTI:* The format of consumed CTI ranges from text files, PDF files, and articles to MISP, STIX, JSON, XML, cvs, and txt files. Of the 30 consumers of CTI, the median of the amount of consumed CTI without interrelationships within the data is 80%. The corresponding median for the amount of consumed CTI with interrelationships within the data is 20%. The sectors or the sizes of the organizations do not seem to influence the results.

Fig. 5. The size of the organizations represented.

Fig. 6. Respondents by sector.

- *Storing and using the consumed CTI*: Of the respondents, 83% replied that they partially or fully store the received data in a structured way, and 37% replied that they use the consumed CTI for further analysis.
- *Obstacles for sharing*: The **Traffic Light Protocol (TLP)** [14] is used for handling and sharing CTI. TLP describes four levels (white, green, amber, red) that dictate how labeled information must be protected and to whom the information may be distributed. Knowledge of TLP is limited outside the technical community and may hence be problematic to use in communities where TLP is uncommon.

In our results, 77% of respondents state that the most prominent reasons for not sharing CTI are related to privacy, confidentiality, and classification of data. Out of these, TLP is mentioned by 26% directly. In addition, 30% refer to classification in general, without specifying a specific classification scheme. For 43% of the respondents, the lack of time, motivation, and resources are the most prominent reason for not sharing CTI.


```

Object:
  "type" = "threat-actor"
  "name" = "sad panda"
  "description" =
    "uses 123.456.789.1 for c&c."
  "object refs":
    "indicator-123"
    "attack-pattern-123"

  "type" = "indicator"
  "id" = "indicator123"
  "name" = "IPAddress=123.456.789.1"

  "type" = "attack-pattern"
  "id" = "attack-pattern-123"
  "name" =
    "c&c attack pattern used by sad panda"

Object:
  "type" = "campaign"
  "name" = "sad panda"
  "description" =
    "sad panda uses 123.456.789.1 for c&c."

Object:
  "type" = "threat-actor"
  "name" = "sad panda"
  "description" =
    "uses 123.456.789.1 for c&c."

  "type" = "indicator"
  "name" = "IPAddress 123.456.789.1"

  "type" = "attack-pattern"
  "name" =
    "c&c attack pattern used by sad panda"

```

Fig. 7. Three different STIX representations of the same piece of information encoded by three different threat intelligence analysts.

4.1.2 Using STIX. Of the respondents, 70% replied that they have not used STIX in the past 6 months. Of the 30% who do use STIX, only 66% have created a STIX file/bundle. Of those who responded that sharing of CTI was part of their role, 80% had not created a STIX file/bundle in the past 6 months. The questionnaire results gave little insight into which SDOs and relationship types were in use due to the low number of respondents.

There is a limited amount of STIX feeds available [27]. The available feeds evaluated by Polzunov and Abraham [27] indicate that there exists “good” and “bad” use of STIX, and they provide a rich set of metrics for evaluating this. Their analysis implies that there is a large span in the number of object types in use, and that the more objects types are utilized, the more custom fields are included as well. The same is true for the use of relationships. We suggest that utilizing more object and relationship types implies advanced use, and that need of custom fields implies shortcomings in the standard. The results then indicate that the more advanced STIX usage becomes, the more shortcomings emerge.

4.2 Discussion

4.2.1 Using STIX in CTI Processes. STIX represents one of the most thorough standards for describing CTI. Yet with the rapid developments in the field of CTI, the standard as it currently stands suffers from limitations. First, the absence of a top-level element to represent and structure specific company assets such as IT systems affected by an incident is a limitation [5]. STIX relationships are flexible and without restrictions for usage. Hence, the same relationship may be used between different SDOs. Since relationships are described within the properties of one object, this gives a potential for confusion in using them. According to Polzunov and Abraham [27], relationships are only used to a limited extent.

A fast-growing knowledge base for CTI is the ATT&CK framework supplied by MITRE [23], which describes threat actors, tools, techniques, and tactics. This knowledge base is a major step toward structuring knowledge of tactical CTI in terms of the TTPs described previously. However, the use of STIX to represent the content of ATT&CK is not straightforward, due to tactics not being represented in STIX as a term. MITRE publishes their ATT&CK knowledge base with the use of STIX, but they have had to add custom fields within the “Attack Pattern” SDO to include all the information. The lack of explicit options to express this relevant knowledge base within CTI is a shortcoming in STIX.

Second, the large flexibility of the STIX language is by itself a weakness. An example piece of information typically shared is “Sad Panda has used 123.456.789.1 for command and control.” When asked to represent this with STIX 2.0, three different threat intelligence analysts came up with three different representations, and we cannot exclude the possibility that additional different representations would be suggested if additional practitioners were asked. Figure 7 presents the simplified code for the three representations.

The three analysts all place information in description fields using English prose rather than structured information consumable by a computer. In addition, there are different object types in use, which means that someone

consuming this CTI has to look several places to ensure consuming everything. Due to the different ways of representing the same information, the possibility of automatic consumption and computer-based analysis is limited. If a computer cannot identify information because the information type is not normalized, then “Big Data” style analysis is not possible. As a result, large amounts of manual work are needed to interpret, correct, and analyze the data. Furthermore, different ways of representing the same information, whether it is consumed manually or automatically, will result in loss due to the normal behavior of both humans and computers to look for what is known, then discard the rest.

These limitations of STIX have emerged as a result of new requirements for precise CTI representation, which seem to be in conflict with the flexibility that had to be included in the early versions of STIX for practitioners to use it. However, as the maturity and capabilities for precise CTI are increasing, we are in need of a more strict and precise model than what STIX currently offers.

4.2.2 CTI in Theory and Practice. Shared data and information may be used to perform or extract CTI but do not always classify as such. NATO and Gartner are aligned in defining CTI as produced knowledge, which cannot be transferred using stand-alone data points. If CTI is shared using only stand-alone data points, vital knowledge regarding the threat actors and the context gets lost in the process, because the interrelationships and context represent the knowledge. The results of the questionnaire indicate that many professionals share CTI, but when digging deeper into formats and actual shared data, it is found that what is shared is simply threat data, and in some cases threat information. The concept of CTI loses its meaning when used for sharing of threat data. As it currently stands, if a practitioner claims to send CTI through STIX, one cannot trust that one will receive more than lists of indicators with no relationships that can only classify as threat data.

4.2.3 What Does It Mean to “Use STIX”? The results from the questionnaire show that although many claim to use STIX, few actually create STIX bundles. Although one does not have to create bundles to use STIX, creating STIX bundles is what defines content and what allows for specifications of how STIX is used. To “use STIX” in terms of consumption of STIX bundles should involve more than accepting a JSON file and manually consuming it. This is because the same content may then just as well be shared as English prose using a text file, PDF, emails, or a csv file. The potential benefits of using STIX are to structure information and create a standardized way of sharing CTI suitable for automation. To “use STIX” would then arguably entail fulfilling at least one of the two, and preferably both. The results reported in Section 4.2.1 show how the usage of different SDOs and relationships varies, which entails that true structure is generally not in place. There are typically different representations of the same CTI, and hence the value of structure is degraded. In addition, also shown in Section 4.2.1, not all information can be represented with the default SDOs as they are currently defined, and hence custom fields and properties must be used. This entails a lack of standardization of all relevant information, which poses a problem for automation. We therefore argue that although many claim to “use STIX,” in most cases it is not used as a standardized way of sharing CTI suitable for automation, even when a STIX bundle or file has been created.

4.2.4 What Value Does STIX Have If Not Used as a Strict Format? If a standardized format is not used consistently, its value is limited. This especially holds for IT applications where standards are essential for distributed applications. The value of representing data or information in a standardized format includes knowing where in a file a certain type of data is found, and what it looks like. If a standard provides flexibility about where to place a certain piece of data, or its format, it reduces the ability of other parties to identify and use this piece of data. If a standard allows ad hoc extensions of representation, the extended parts of the standard requires additional work for other parties to consume. In both cases, the standard will significantly reduce its value for computers and human operators using it for sharing.

We argue that STIX currently has this deficiency. This means that the current usage of STIX is not superior to any other standardized way of sharing data that two or more parties have agreed on, based partly on a common vocabulary. The STIX vocabularies are valuable contributions. If CTI is shared in a format that the recipient

does not understand, the recipient's ability to consume the knowledge is limited. The labor-intensive task of agreeing on how to use the standard between two parties could solve this. If no agreement between two parties has been made, or a parsing task has not been conducted on the receiving end, then there will be a significant loss of information in transfer. Most importantly, if the standards are not used consistently, the threat intelligence community's ability to know if they are talking about the same threats/information is lost. Standardized sharing is key for CTI, and for STIX to be useful it needs to be used in line with intent.

4.2.5 The Issue with Claimed Usage of Sharing Standards—When Ideals Do Not Match Reality. Through examination of one of the “known truths” in CTI, this study finds that assumptions made regarding the use of STIX are not valid. The three main consequences these types of assumptions lead to are all results of using “known truths” as guidance for prioritizing the work and development within a field. Prioritizing a task or a fact means deprioritizing something else. First, we find that training personnel to use STIX as it stands today takes valuable time away from other types of training that can potentially hold more value. This influences the shortage of technical security personnel [12, 33] in a negative direction as personnel may be less capable and less efficient to do the required work. Second, choice and development of tools and procedures need to adhere to the reality. The field of CTI is highly dependent on technical tools and solutions, and the effectiveness and capabilities of the collective workforce rely on informed choices. Priorities based on imprecise information can lead to decreased effectiveness and capability. Last, research and development needs to focus on real-world problems and to prioritize based upon a rational foundation. Assumptions and “known truths” need validation, and if an assumption is found to be wrong, that fact must be disseminated to the community. This validation is necessary to steer ongoing research in a direction that can benefit our collective cyber defenses. Although the research presented here is a micro study, it touches on critical foundations of the processes of CTI and methods within the field. The ramifications identified can extend to international cooperation. Agreements for multinational cooperation to improve cybersecurity beyond country borders for a secure international cyberspace can be made on an international level [1]. However, if cooperation on the technical level is hindered, the potential for collaboration at a political level is limited. In the current landscape of cyber threats, it is essential to have a common language for sharing and acting on CTI through public and private efforts with the aim to secure cyberspace. Without this, the ability to develop automated tools that are able to utilize the CTI is restricted, and the global community's ability to defend against threats in and from cyberspace is unnecessarily hindered.

5 THE DATA MODEL

As a suggestion for structuring CTI to facilitate automation, we have created a data model to meet the requirements set in Section 1.1 and implemented it using a backend based on Apache Cassandra¹¹ and Elasticsearch.¹² We have implemented an Apache TinkerPop¹³ graph engine that enables graph querying with the use of the graph query language Gremlin.¹⁴

Note that a graph view is not the same as a graph database. You can display any kind of data, even a flat text file, as a graph, but you cannot use graph queries unless you have a graph engine interfacing with your data.

An implementation of our data model can be found on GitHub¹⁵ under the ISC license. An openly available instance of the same implementation can be found online.¹⁶

This section is divided into four parts: the foundation of our data model and the discussion leading to it, the schema design, the choice of allowing placeholder objects, and the evaluation of the data model.

¹¹<http://cassandra.apache.org/>.

¹²<https://www.elastic.co/>.

¹³<http://tinkerpop.apache.org/>.

¹⁴<https://tinkerpop.apache.org/gremlin.html>.

¹⁵<https://github.com/mnemonic-no/act-platform>.

¹⁶<https://act-eu1.mnemonic.no/>.

Fig. 8. Objects and fact.

5.1 Foundation: Objects and Facts

The foundation for our work has been a data model consisting of objects and facts. We can define different object types and different fact types. Thinking of graphs, objects are the vertices and facts are the edges. Objects can be described as nodes, and facts may be described as relationships. In the following, we use the terms *objects* and *facts* as illustrated in Figure 8.

The specifications and restrictions to this model is given in the following.

5.1.1 Immutable Facts. Objects are defined globally. There are no properties except the object value itself linked to an object, and everything you know about one object is stored as facts. All data included in the data model must be included by adding facts. A fact may connect to one (object property) or two (relationship) objects that will be created when they are pointed to by a fact. A fact is directed and can be bidirectional.

All facts are timestamped and immutable. Removing or altering a fact is hence not possible; however, a new fact can be added that retracts the old one. In this way, we make sure nothing is deleted and can prevent repudiation. This way, we also preserve history and check the history of the dataset.

All facts have an owner that allows for multi-tenancy.

5.1.2 Time. Because facts are timestamped and cannot be deleted, we are able to traverse the available data back and forth in time. Using the available threat intelligence in an incident response setting, this is useful for two reasons.

First, we can see exactly what we knew at a given point in time. In situations where a range of decisions are made within a time frame of months, it is useful to be able to turn back time to see what information was available at the time the decision was made. When incorrect decisions have been made, the ability to go back in time and see what information was available at that time will provide the ability to learn from mistakes.

Second, we can see how a threat has evolved over time. To know what infrastructure, behavior, and resources a given threat actor has used at different times is useful to separate threat actors from each other, to identify copycats or impersonation, and to evaluate how advanced the threat actor is. A threat actor using novel techniques but abandoning them when they become normal behavior may be considered more advanced than others.

5.1.3 Trust and Confidence. Given the nature of CTI, where large amounts of data, information, and knowledge are shared and collected without always being true or accurate, an analyst needs evaluation support. An evaluation of how much confidence should be assigned to a given piece of CTI is often a calculation on how much we trust the origin of the CTI, and how certain that origin claims to be about the CTI. This is solved in our data model on each fact as follows. Every added fact has an origin. Every origin is assigned a trust score between 0 and 1. All facts are assigned a confidence score between 0 and 1. The product of the trust and confidence scores for a fact gives the analyst a certainty score that provides decision support.

5.1.4 Access Control. When sharing CTI, and having different people accessing the data, we need to have proper access control in place. We want to make sure we are able to share despite different sources, different confidentiality requirements, and schemes such as TLP labeling [14]. This was a major part of the motivation for creating our data model.

The solution to this has been to have granular access control on facts, both role based and explicit. If you have access to a fact that is related to an object, then you have access to that object as well. If you do not have access to any facts related to an object, then access to this object is refused. Note that this will influence results of analysis by graph traversal in an important way. The facts are used as edges in the graph, and traversal of

Table 1. Sources Influencing the Data Model

Source	Relevant Object Types
MITRE ATT&CK (consumed the published STIX representation of their knowledge base)	Tactic, Technique, Tool, Threat Actor
VirusTotal	IPv4, IPv6, FQDN, URI, Content, Hash, Tool, ToolType, Query, Path (scheme and base-name as facts)
Shadowserver ASN	IPv4, IPv4Network, ASN, ¹⁷ Organization, Country
Passive DNS	IPv4, IPv6, FQDN
The VERIS Community Database (VCDB)	Incident, Tool, Threat Actor, Organization, Country, Campaign, Sector
MISP Galaxies	Tool, Threat Actor, Sector
STIX vocabularies	Sector
Open Source Intelligence extracted with Natural Language Processing	All

the graph will depend on the available facts. In certain cases, a single fact may represent the only connection between two subgraphs, and taking away access to this fact may have a large impact on results.

5.2 The ACT Data Model

Based on our object/fact foundation, we have defined a set of object types and fact types that are relevant and necessary for our domain. The initial selection of object types was influenced by STIX [3], the Detection Maturity Model [32], and the Diamond Model [9], as well as available Open Source Intelligence extracted with the use of Natural Language Processing and our own experience. Fact types were added as we found them useful, with an increasing attention to the semantics and the characteristics of each of them. As our use cases for querying the data expanded, we saw the usefulness of differentiating between fact types. Figure 9 shows the complete data model schema as a graph. The diamond shapes represent the values of fact types connected to only one object type.

We have populated the data model with a range of sources. A list of openly available sources used for improvement and validation so far can be found in Table 1. The data model has been developed and improved along with introduction of new data.

In the following, we explain the background and reasoning for the choices we have made and include results from importing different data sources.

5.2.1 Consistent Data. There are restrictions on which fact types can be used to link which object types as seen in Figure 9. These restrictions enforce data consistency by preventing different representations of the same data, which is a known problem with current attempts to model CTI. If we allow flexibility in how different data can be represented and introduce a range of users, then the data quality in terms of format is quickly reduced.

The granularity of the data model is intended to be aligned with pivot points normally used by analysts. This has been achieved by covering all sources in the currently available models and sources of CTI, described in the introduction to this section.

We started with differentiation between “malware,” “tool,” and “utility,” all instances of software. However, we saw that the definitions of the different terms varied in different sources and became difficult to maintain. This is consistent with the known problem of classifying malware, and we do not attempt to solve this in our data model. When the content of the CTI was not consistent, we found that there was no value of using it. Therefore,

Fig. 9. The complete data model represented as a graph.

in our platform, these concepts were all rolled up into “tool,” with the possibility of tagging them as “malware” or “utility” as appropriate.

5.2.2 Enrichment and Query/Analysis across Sources. One of our first observations was that our graph ended up being a series of subgraphs, and we wanted to be able to connect them. The simple solution was enrichment. As we added more enrichment sources, the graph gradually became more and more interconnected, and we could find new connections between clusters of information that were originally separate.

Pivoting on an object is useful, as it lets you find related information and give you a more comprehensive context. One simple example is from DNS: start with a domain name, find all of the IP addresses that it has resolved to, and then find all other domain names that have resolved to those IP addresses.

Passive DNS data is a historic record of DNS lookup resolutions that is important for an investigation. Since 2013, a mnemonic has collected pDNS data. By 2017, when we had the initial version of the platform ready for data consumption, we had a TLP: a white dataset of approximately 100 GB of data. By analyzing super nodes in the dataset, we have discovered new and unknown sinkholes. We tag known sinkholes with a fact connecting to the object to filter them out when traversing the graph further.

A more advanced solution was to use classifiers to bridge technical, tactical, operational, and strategic threat intelligence. An example of this is using VirusTotal to bridge technical indicators to tactical information in MITRE ATT&CK. We extracted the malware family name from antivirus signatures and normalized it. We then normalized the software entries from MITRE ATT&CK (e.g., “TrickBot” became “trickbot”). Automated enrichment

Fig. 10. An example of fact types connecting to and from an URI object, composing the total URI.

with VirusTotal then connects file hashes and network infrastructure to the “trickbot” object that is again linked to the tactical threat intelligence in ATT&CK.

We also observed that we could create uncommon pivot points, and our URI object type is an example of this. A URI object is just a UUID connecting different components to each other for a complete URI. Figure 10 shows the facts connecting to a URI. Given a URL, we split it into the host (domain/IP) part, the path, and the query parameters. Pivoting on query parameters proved useful when tracking spam campaigns with specific phishing kits, as all of the other pivot points changed for each spam run, but the query parameter stayed the same.

5.2.3 What Is Content? The concept of “content” is an example of where we need to be precise in order to enable automation. In the context of CTI, we handle not just files but also stream segments, text strings, and parts of content that has been found in memory. This is all “content,” but should not all be classified as files. Furthermore, even in the case of a file, we find that it is seen as unique based on more than one property. We argue that the filename, the actual content, and the location of the content together is what we refer to when we describe something as a unique file.

To illustrate the preceding, we use the example of two files with the file system path `/etc/hosts` on two different Linux machines. In a given situation, the name and content may be the same, but they are still not the same file since they reside on different machines. In a different scenario, one can find two files with the same name on the same machine but with completely different content. In both cases, everyone agrees on the files being different from each other.

To be able to describe these things in a precise manner, and to identify similarities and identical objects, we saw the need for splitting them. The result was “content” linked to “uri” with the fact types as seen in Figure 11. The basename (which includes the filename) is included within the URI.

The “at” fact found connecting a content object to a “uri” object is in the meaning of “seen at” and “downloaded from.” The general “at” was selected so as to not exclude any of the terms. The additional “connectsTo” fact represents a content that has been seen connecting to a “uri.” The two fact types illustrate the very different scenarios where there is a link between the two object types. This is an example of the importance of semantics when handling CTI.

Fig. 11. An example of fact types connecting to and from a content object.

5.2.4 Aliasing. Our data model allows for aliasing different names for the same object.

Instead of giving a threat actor a primary name, like in MISP Galaxy, we use “alias” as a fact type between threat actor names that are known or suggested to be the same. The same is done for tools. This may be seen in Figure 9. Adding information on any threat actor’s name is then done by linking to the name given at the source, hence no “main name” is used. In this way, if an alias turns out to be wrong, you only need to retract that one alias fact, and the rest of your information is still correct.

The problem of different names for the same object is a common situation in CTI. Often, we see that different providers of CTI assign their own primary name for the object and connect all information about this object to that name. For instance, if selecting “APT28” as the main name for a threat actor, and receive information about “Fancy Bear” (an alias for APT28), then such a solution will connect the information to “APT28.” This information can of course be wrong. If you at some point in the future decide that “Fancy Bear” is not an alias for “APT28,” then you would have a large manual task in correcting your data.

The “alias” fact type is used between threat actors and tools and might be applied to other object types in the future.

5.3 Placeholders to Preserve Information

In the ACT data model, you cannot link objects without a defined fact type between the object types. From adding new sources in various structures and formats, we found ourselves in need of adding more fact types based solely upon the information we wanted to consume. This resulted in a vast amount of fact types and no consistency in representation of information. This is one of the mentioned weaknesses of the structure given in STIX, where there are several ways of representing the same CTI, resulting in problems digesting all information, especially without manual work and deduplication.

Looking for solutions, we found the need for describing *things we know exist, but know little about*. Using blank nodes has been a solution for this problem in the field of ontologies [17] and is part of the standardized W3C RDF Semantics [35]. We introduced the same thought in our data model, by using what we called *placeholders*. The idea is that the user may find information about the object in the future, then replace the placeholder with an actual object through a new fact. In this way, we were able to strictly define how the data are truly connected to each other, without worrying about having all data in a chain to consume it.

As an example, Figures 12 through 14 illustrates a typical scenario when working with CTI.

After implementing placeholders in our data model and restricting the fact types’ possible connections, we found that adding and searching the data gave us an easy overview over what data is missing. This is a very

Fig. 12. A typical piece of information received as CTI.

Fig. 13. The information needed to give the statement in Figure 12.

Fig. 14. The need for placeholders. Information we know exists in gray but is often not available for sharing.

interesting benefit for security analysts receiving or searching data on a relevant incident, both to know what data you do not have, but also to know what data others will need to be in possession of when sending you data. When judging the usefulness of different CTI sources, this is a relevant analysis to perform.

5.4 Evaluating the Data Model

The data model proposed in this study is an ontology. When evaluating the data model, we have therefore looked to the ontology literature and used the guidance found in other works [13, 20, 25].

McDaniel and Storey [20] describe different approaches to ontology evaluation based upon published assessment efforts. Degbelo [13] explains that there is a recognizable gap between the theory and practice of ontology evaluation. Apart from expressiveness and practical usefulness, very few of the criteria suggested by previous work have been often used.

The approach referred to by Obrst et al. [25] as evaluation with respect to domain data sources is well aligned with our development process. This may be considered part of evaluating the expressiveness of an ontology. Further, the objectives of the data model as described in Section 1.1 have been the evaluation criteria used throughout our iterative development process, which have been included as well.

The theoretical publications on ontology evaluation criteria gives us a range of possible additional metrics for evaluating our ontology/data model. We have added semantic agreement and reasoning capabilities to complete our data model evaluation.

The following describes the results of the evaluation per evaluation criteria.

5.4.1 Practical Usefulness. The data model has been developed based upon the data sources found within the CTI domain. In this way, the data model is well aligned with other ontologies covering the same field (the ontologies lying as a basis for the data sources given in Table 1). All identified sources can be consumed without loss of information. Creating export and import functions to and from an implementation of the data model is not problematic and is partly done and published on GitHub.

As mentioned in Section 3.1, there is a certain level of understanding required to ingest new data due to the strictness of the data model. For manual inputs, our implementation of the data model uses the graphical user interface to help the user. For automatic ingestion from new sources of CTI, an understanding of the data model and the published API is required. The data model is well documented, and the benefits from a strict data model are far exceeding the benefits of adding data in random manners. Implementation is not done in a standardized ontology language. However, there is no reason our data model cannot be expressed in, for example, RDF and OWL if needed.

The implementation of our data model is done with a ISC license, making it available for everyone. The documentation of the data model is available at GitHub.

5.4.2 Expressiveness. Expressiveness is evaluated with respect to the data sources we used for populating the data model as described in Table 1. When publishing this work, we have reached a point where no new sources of data to include in the data model have been found that cannot be ingested without loss of information or knowledge. The data model is considered stable in the sense that new development of the data model are changes that increase the amount of knowledge found in the data model, beyond the information or knowledge the original data sources could provide.

5.4.3 Consistent Data. Data quality is both format and content of the data. The quality of content from different sources is beyond the scope of improving by providing a data model, but expressing the context of all input increases the understanding of this content. As such, data quality in terms of content is not reduced when using the data model. The format of the content is strictly enforced in the provided data model. All CTI must be consumed as facts—no objects are allowed directly. There is no possibility of adding the same data in different ways, as there is no use of open fields. This ensures consistency of data: the content found in one place will always represent the same thing, and looking for a certain thing can always be done in completeness in one place. As an example, the three different solutions for representing the same content shown in Figure 7 would not be possible using our proposed data model. There are no description fields available for English prose, and all parts of the content are covered in the range of object and fact types. The producing analyst is forced to identify the needed object and fact types to represent the desired content. All inconsistencies or vocabulary differences will hence be handled on ingress. In this way, all consumed CTI will be represented in a consistent structure, and the analyst querying the CTI can take advantage of the restrictions enforced in the data model—less knowledge is needed for extracting the desired CTI. Creating a data model that enforces consistent data provides increased automation possibilities.

5.4.4 Semantic Agreement and Consensus Building. The motivation for creating the data model was not to solve the issues of differences in use and understanding of the terms found within CTI. Hence, the data model does not solve this. However, a motivation was to make sure we do not add to the confusion. The data model does contribute positively as it forces the users to be consistent in the use of terms, and it explains the terms by detailing the context of how the different concepts are related to other concepts. In situations where the different data sources showed differences in use of the terms, the data model chooses to generalize in order to not exclude CTI and not decrease the value of the knowledge graph as a whole. Examples are “tool” and “content” as described in Sections 5.2.1 and 5.2.3. The use of “alias” reduces the impact of mistakes in the different sources.

Further, as described under expressiveness, the current data model covers the concepts found in the identified sources. We have added suitable vocabularies where this is natural.

5.4.5 Reasoning Capabilities. Enforcing triplets is enforcing graphs, which is then enabling all graph-related analysis techniques. The definition of *factTypes* and which *objectTypes* can be connected with them are restrictions that express knowledge of the CTI domain. These restrictions are used for enhancing the value of the added CTI and improving the graph query reasoning capabilities.

The fact that the data model is not expressed in a standardized ontology language is limiting the use of known ontology tools for further development, reasoning, and formal verification of the completeness and consistency. An RDF/RDFS/OWL implementation may provide more possibilities and is planned for future research as a natural way forward.

5.5 Discussion

The data model that we propose is strict: it restricts which relationships may be added to connect two objects, and it enforces that objects may not be added directly but through facts. The main benefit from this is a consistent dataset that enables automation and improves data quality. It reduces the computational load of graph queries as the range of possible paths in the graph is restricted. It also provides for easier graph queries as there is no need to know the data you query so long as the user understands the data model. As an example, there is a limited amount of traversals of the graph between “threat actor” and “technique.” Knowing this makes it trivial to find all connections between known “threat actors” and the “techniques” we know it has used without missing any available data. With this, we argue that building the data model has transferred some of the advanced knowledge from CTI professionals into the model itself, which enables less skilled professionals to analyze the same data with consistent results.

In the course of developing this model, we have discussed various solutions, implemented them, and have been positively surprised by some of the findings. The following discourse will look into the most relevant insights.

5.5.1 Data Validation. Large datasets often include some data that do not comply with the given specification. Adding data to our model, outside specifications, will be identified fast as it will fail upon consumption.

Inconsistent or incorrect CTI may lead to bad results because it may cause incorrect conclusions and because they may disturb some of the other data. Most times, errors occur due to mistakes entered at the source because of the complexity of the subject matter or because multiple authors use different methods or terminologies.

We have found that the model allows for data validation. As an example, when querying the data from ATT&CK using our data model, we found that there actually was one technique called *Shared Webroot* without a link to any threat actors or any tools, which in threat intelligence is an interesting observation. Knowing that ATT&CK only includes data they have a reported observation of means that this technique has been observed, but not described, by openly available sources. This was obvious when we applied our model.

Adding MISP Galaxy for threat actors¹⁸ where there is a range of users adding data with limited restrictions on data inclusion, we found that all threat actors were listed under a main name, with all information about them linking to this name. There are aliases listed underneath, but with no capability of reasoning on these aliases, the result is that a large portion of the threat actors actually are connected and seen as one. This meant that the value of the information was diluted as almost all information known about one threat actor was also stated to be valid for a large amount of other threat actors. This is an example of validation that may be used for evaluation of CTI sources, and it shows the importance of the chosen solution of aliasing as chosen in our model.

5.5.2 Evaluation of CTI Sources. When evaluating different sources of CTI, it is useful to evaluate the quality of the offered data. Our data model may be used for this purpose. First, by adding context and knowledge to your data, which enables interpretation of the received data. Extensive aliasing, wrongful classifications, or attributions may be easily found through such evaluation. Second, it helps with finding data with errors, inconsistencies, or bad formatting. The strictness of the data model excludes the possibility of importing data with

¹⁸<https://github.com/MISP/misp-galaxy/blob/master/clusters/threat-actor.json>.

errors, inconsistencies, or bad formatting. When working to include new data sources, these shortcomings will surface. Third, the data model can be used to check what data is missing. When utilizing the data model with a given dataset, if there is missing data, it can be identified by identifying cases of missing data in between data points. We can also find what object and fact types are used in that data to evaluate the range of CTI provided from the source.

5.5.3 Agreeing on Terms and Relationships. The terms and concepts within CTI are often referred to with different meaning. An example of this is “campaign,” which often is used to describe stand-alone incidents and relevant threat actors in addition to the collection of incidents by the same threat actor targeting a given sector or geographical location. When connecting each concept to other concepts in a defined way, the data is given context and thereby provides additional meaning to a user. In this way, we argue that ambiguity in terms and definitions will be reduced.

5.5.4 Differences in Object Types and Fact Types. There is a difference between objects that may be observed directly and objects that are a result of human decision or analysis. Example of these types are “incident” and “tool” (not “content” or “hash”). The relationships going to and from these may also imply analysis, like “classifiedAs” and “attributedTo.” These facts are not a directly observable link. The trust we have in the source of these facts is thus more significant.

The differences in meaning of the different fact types indicates the importance of semantics. There are object types that have multiple possible fact types connecting them, and where the semantics of the chosen fact type significantly differentiates.

An example of this is *content* $\xrightarrow{\text{connectsTo}}$ *URI* and *content* $\xrightarrow{\text{at}}$ *URI* as described in Section 5.2.3.

5.5.5 Sharing CTI. Newer publications suggest that still about 78% of shared CTI is unstructured [29]. Without any structure, we can only automate sharing of *data*, as no relationships are present. With the choice of only adding information as facts (relationships) in ACT, we force all CTI to be stored with/as relationships. With this baseline, we can automate sharing of triplets, which is a significant improvement from sharing data and allows for sharing of graphs.

6 CONCLUSION

Through questioning the difference among data, information, and intelligence in models and standards used in CTI, this article finds that although sharing threat intelligence is deemed to be crucial, classification and trust, unclear use of terminology, and large flexibility within STIX hinder developments in the field of CTI. Although the current flexibility of STIX has allowed for inclusion of a variety of users, the lack of precision reduces the possibility for knowledge transfer and data analytics. To improve this situation, stricter definitions and greater specificity are called for. Increased precision and clearer guidelines can enable a full use of STIX without loss of vital information, which in turn can create possibilities to share knowledge beyond flat files. Such a use of CTI can improve the ability to defend collectively in the cyber domain. Although the international community calls for shared efforts to secure cyberspace, progress will be limited if the technical foundation is not in place to do so.

We have proposed a strict data model based on objects and relationships, with the ability to represent available CTI. We have populated it with relevant data, and have identified new information through analysis enabled by the data model. The most prominent results from the data model are data validation, seamless enrichment, excellent analysis capabilities, and flexibility of CTI ingest. When new sources of CTI emerge, we will use them to validate the data model.

Future development of the data model will include hierarchical object types and fact types (using relationships borrowed from ontologies like “subClassOf” and “subPropertyOf”), which will enable inheritance, more precision, and reasoning.

In the implementation of our data model, we allow external programs to access the content and add new facts. In this context, we are exploring the use of an OWL-implemented version of our data model to infer new facts based on rule-based reasoning using Semantic Web Rule Language (SWRL) [34].

ACKNOWLEDGMENTS

To the anonymous reviewers: thank you for valuable input.

REFERENCES

- [1] NATO Standard AJP. 2016. 2.0 Allied joint doctrine for intelligence, counterintelligence and security doctrine.
- [2] ANSSI, Lutix, and CERT-EU. 2019. The OpenCTI Platform. Retrieved August 7, 2021 from <https://github.com/OpenCTI-Platform>.
- [3] Sean Barnum. 2012. *Standardizing Cyber Threat Intelligence Information with the Structured Threat Information Expression (STIX™)*. Technical Paper. MITRE Corporation.
- [4] David Bianco. 2014. The Pyramid of Pain. Retrieved August 7, 2021 from <http://detect-respond.blogspot.no/2013/03/the-pyramid-of-pain.html>.
- [5] Fabian Böhm, Florian Menges, and Günther Pernul. 2018. Graph-based visual analytics for cyber threat intelligence. *Cybersecurity* 1, 1 (2018), 16.
- [6] Siri Bromander. 2019. Questionnaire: Sharing Cyber Threat Intelligence. Retrieved August 7, 2021 from <https://github.com/sbrom/sharingCTI/>.
- [7] Siri Bromander, Audun Jøsang, and Martin Eian. 2016. Semantic cyberthreat modelling. In *Proceedings of the 11th International Conference on Semantic Technology for Intelligence, Defense, and Security (STIDS'16)*. 74–78.
- [8] Siri Bromander, Lilly Pijenburg Muller, Martin Eian, and Audun Jøsang. 2020. Examining the “known truths” in cyber threat intelligence—The case of STIX. In *Proceedings of the International Conference on Cyber Warfare and Security*. 493–XII.
- [9] Sergio Caltagirone, Andrew Pendergast, and Christopher Betz. 2013. *The Diamond Model of Intrusion Analysis*. Technical Report. DTIC Document.
- [10] D. Chismon and M. Ruks. 2015. *Threat Intelligence: Collecting, Analysing, Evaluating*. Technical Report. MWR InfoSecurity, London, UK.
- [11] European Commission. 2019. What Is an SME? Retrieved August 7, 2021 from https://ec.europa.eu/growth/smes/business-friendly-environment/sme-definition_en.
- [12] William Crumpler and James A. Lewis. 2019. The Cybersecurity Workforce Gap. Retrieved August 7, 2021 from https://csis-website-prod.s3.amazonaws.com/s3fs-public/publication/190129_Crumpler_Cybersecurity_FINAL.pdf.
- [13] Auriol Degbelo. 2017. A snapshot of ontology evaluation criteria and strategies. In *Proceedings of the 13th International Conference on Semantic Systems*. 1–8.
- [14] FIRST ORG. 2019. Traffic Light Protocol (TLP). Retrieved August 7, 2021 from <https://www.first.org/tlp/>.
- [15] Gartner. 2013. Definition: Threat Intelligence. Retrieved August 7, 2021 from <https://www.gartner.com/en/documents/2487216/definition-threat-intelligence>.
- [16] S&P Global. 2018. Global Industry Classification Standard. Retrieved August 7, 2021 from https://www.spglobal.com/marketintelligence/en/documents/112727-gics-mapbook_2018_v3_letter_digitalspreads.pdf.
- [17] Aidan Hogan, Marcelo Arenas, Alejandro Mallea, and Axel Polleres. 2014. Everything you always wanted to know about blank nodes. *Journal of Web Semantics* 27 (2014), 42–69.
- [18] Jon A. Krosnick. 2018. Questionnaire design. In *The Palgrave Handbook of Survey Research*. Springer, 439–455.
- [19] Vasileios Mavroeidis and Siri Bromander. 2017. 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'17)*. IEEE, Los Alamitos, CA, 91–98.
- [20] Melinda McDaniel and Veda C. Storey. 2019. Evaluating domain ontologies: Clarification, classification, and challenges. *ACM Computing Surveys* 52, 4 (2019), 1–44.
- [21] Florian Menges, Christine Sperl, and Günther Pernul. 2019. Unifying cyber threat intelligence. In *Proceedings of the International Conference on Trust and Privacy in Digital Business*. 161–175.
- [22] MISP. 2019. The MISP Platform. Retrieved August 7, 2021 from <https://github.com/MISP/MISP>.
- [23] MITRE. 2019. Adversarial Tactics, Techniques and Common Knowledge (ATT&CK). Retrieved August 7, 2021 from <https://attack.mitre.org/>.
- [24] OASIS CTI TC. 2017. Structured Threat Information Expression (STIX™) 2.0. Retrieved August 7, 2021 from <https://oasis-open.github.io/cti-documentation/>.
- [25] Leo Obrst, Werner Ceusters, Inderjeet Mani, Steve Ray, and Barry Smith. 2007. The evaluation of ontologies. In *Semantic Web*. Springer, 139–158.
- [26] University of Oslo. 2019. Nettskjema. Retrieved August 7, 2021 from <https://www.uio.no/english/services/it/adm-services/nettskjema/>.

- [27] Sergey Polzunov and Jorg Abraham. 2019. Evaluate or Die Trying: A Methodology for Qualitative Evaluation of Cyber Threat Intelligence Feeds. Retrieved August 7, 2021 from <https://www.first.org/resources/papers/london2019/EVALUATE-OR-DIE-TRYING-Abraham-Polzunov.pdf>.
- [28] Andrew Ramsdale, Stavros Shiaeles, and Nicholas Kolokotronis. 2020. A comparative analysis of cyber-threat intelligence sources, formats and languages. *Electronics* 9, 5 (2020), 824.
- [29] Clemens Sauerwein, Irdin Pekaric, Michael Felderer, and Ruth Breu. 2019. An analysis and classification of public information security data sources used in research and practice. *Computers & Security* 82 (2019), 140–155.
- [30] Clemens Sauerwein, Christian Sillaber, Andrea Musmann, and Ruth Breu. 2017. Threat intelligence sharing platforms: An exploratory study of software vendors and research perspectives.
- [31] Jolene D. Smyth, Don A. Dillman, Leah Melani Christian, and Michael J. Stern. 2006. Comparing check-all and forced-choice question formats in web surveys. *Public Opinion Quarterly* 70, 1 (2006), 66–77.
- [32] Ryan Stillions. 2014. The DML Model. Retrieved August 7, 2021 from http://ryanstillions.blogspot.com/2014/04/the-dml-model_21.html.
- [33] Rebecca Vogel et al. 2016. Closing the cybersecurity skills gap. *Salus Journal* 4, 2 (2016), 32–46.
- [34] W3C. 2004. Semantic Web Rule Language. Retrieved August 7, 2021 from <https://www.w3.org/Submission/SWRL/>.
- [35] W3C. 2014. RDF 1.1 Semantics. Retrieved August 7, 2021 from <https://www.w3.org/TR/rdf11-mt/>.
- [36] Cynthia Wagner, Alexandre Dulaunoy, Gérard Wagener, and Andras Iklody. 2016. MISP: The design and implementation of a collaborative threat intelligence sharing platform. In *Proceedings of the 2016 ACM Workshop on Information Sharing and Collaborative Security*. ACM, New York, NY, 49–56.

Received October 2020; revised December 2020; accepted March 2021