Pahlevan, Maryam; Voulkidis, Artemis; Velivassaki, Terpsichori-Helen

Secure exchange of cyber threat intelligence using TAXII and distributed ledger technologies - Application for electrical power and energy system

Published in:
Proceedings of International Conference on Availability, Reliability and Security, ARES 2021

DOI:
10.1145/3465481.3470476

Published: 17/08/2021

Document Version
Peer reviewed version

Please cite the original version:
Pahlevan, M., Voulkidis, A., & Velivassaki, T-H. (2021). Secure exchange of cyber threat intelligence using TAXII and distributed ledger technologies - Application for electrical power and energy system. In *Proceedings of International Conference on Availability, Reliability and Security, ARES 2021* [122] ACM. https://doi.org/10.1145/3465481.3470476

Secure exchange of cyber threat intelligence using TAXII and distributed ledger technologies - application for electrical power and energy system

MARYAM PAHLEVAN, Aalto University, Finland
ARTEMIS VOULKIDIS, Synelixis Solutions SA, Greece
TERPSICHORI-HELEN VELIVASSAKI, Synelixis Solutions SA, Greece

The energy sector has been, in recent years, the target of sophisticated cyberattacks. Although the importance of collaborative cyber-security consciousness, expressed as extensive cyber threat intelligence sharing, is undoubted, the standardization of the means of exchanging cyber threat information efficiently and securely has been inadequately addressed and is mostly expressed by the emergence of the Trusted Automated eXchange of Indicator Information (TAXII^{TM}) protocol which faces major deficiencies when it comes to data integrity assurance and suitability for event-driven architectures. This paper presents a novel approach enabling secure and real-time exchange of cyber threat information, by extending the technological capacity of the TAXII framework and addressing its deficiencies through the integration of Distributed Ledger Technologies (DLT) and a generalized publish-subscribe middleware. The applicability of the proposed solution has been validated in several use cases addressing the real needs of Electrical Power and Energy Systems.

Additional Key Words and Phrases: **Energy Sector**, **Cyber Threat Intelligence**, **TAXII Framework**, **Distributed Ledger Technology**, **Publish-Subscribe Middleware**

ACM Reference Format:
Maryam Pahlevan, Artemis Voulkidis, and Terpsichori-Helen Velivassaki. 2021. Secure exchange of cyber threat intelligence using TAXII and distributed ledger technologies - application for electrical power and energy system. In *The 16th International Conference on Availability, Reliability and Security (ARES 2021), August 17–20, 2021, Vienna, Austria.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3465481.3470476

1 INTRODUCTION

Critical Infrastructures (CI) may be defined as collections of cyber-physical systems whose disruptions or failures result in destructive financial and health consequences for large population segments [10, 12]. Electrical Power and Energy Systems (EPES) constitute some of the most essential critical infrastructures [45], particularly due to the high dependency of other technologies, services as well as CIs on the proper operation and availability of power networks. The introduction of Information and Communication Technology (ICT) in the operational processes of modern EPES [30], although of great importance, introduces new risks to the operation of EPES, in the form of cyberattacks targeting the energy networks. Under this perspective, EPES protection against cyberattacks has drawn significant attention in the literature [17, 19, 39]. To combat adversaries, conventional information security technologies such as Intrusion Detection Systems (IDS) have emerged as integral parts of the operations of EPES, they are, however, unable to securing EPES infrastructure against advanced/complex cyberattacks [19].

Manuscript submitted to ACM

Several studies tackling modern cyber-security challenges in the energy sector collectively highlight the need for highly scalable and secure cyber-security information sharing platforms (CIS) [13, 23, 42, 44]. At the same time, sector-specific Information Sharing and Analysis Centres (ISACs) such as European Energy–Information Sharing and Analysis Centre (EE-ISAC) and the Electricity Information Sharing and Analysis Center(E-ISAC) have been recently formed to establish trustworthy connections between governmental institutions and relevant industry actors including vendors and utilities [19] and also to enable reliable sharing of Cyber Threat Information (CTI) towards accelerating detection and mitigation of cyberattacks [34].

Despite the definite need for CTI data exchange, little work has been conducted towards standardizing the means of achieving this exchange securely and efficiently. One of the most prominent and widely used frameworks towards that direction is the adoption of the Structured Threat Information eXpression (STIX) [29] data model in combination with Trusted Automated eXchange of Indicator Information (TAXII^{TM}) [6] framework. The latter, however, presents major deficiencies with regards to i) authenticity and non-repudiation auditing of data and ii) suitability for event-driven architectures and real-time applications, even though the protocol already considers relevant operation modes.

This paper presents a novel approach towards enabling secure and real-time exchange of CTI data related to the security status of EPES infrastructures, by extending the technological capacity of the TAXII framework, hence addressing the previously identified deficiencies by properly integrating Distributed Ledger Technologies (DLT) and a generalized publish-subscribe middleware, respectively. More accurately, integration of the TAXII framework and DLTs which are widely used as immutable and append-only data storages, address the need for integrity and audit trail of data. On the other, the publish-subscribe middleware facilitates near real-time exchange of CTI data within the TAXII framework. Eventually, this work verifies the applicability of the proposed solution which enables secure, tamper-proof and highly scalable cyber threat information sharing, by conducting different experiments on the target prototype.

The rest of the paper continues as follows; section 2 provides a brief overview of the discussed technologies. In section 3 a relevant literature overview is presented and followed by a thorough description of the proposed solution in the context of the application of TAXII framework to EPES infrastructures, in section 4. The experimental results are presented in section 5. Section 6 offers a discussion over the presented results. Section 7 concludes the paper.

2 BACKGROUND

2.1 Structured Threat Information eXpression

CTI data exchange is crucial in cyber-security systems since it implicitly enables collaborative cyber-security consciousness. The relevant data is usually shared between either component of common infrastructures or distinct trusted parties through secure communication channels. In an attempt to foster interoperability among different systems producing and consuming CTI data, the STIX ontology was introduced as a community-driven effort, providing standardized, structured representations for threats-related information. STIX is a modelling language that standardizes the representation of CTI and is appropriate in numerous cyber-security application scenarios such as cyber-threats analysis, identifying indicator patterns, coordinating countermeasure activities and, most importantly, sharing CTI data. As a result, STIX is widely deployed by several organizations for reporting and sharing CTI data, as a standardized tool for representing threat information, effectively unlocking the opportunity for different industries to enhance interoperability, efficiency, and consistency of data in the context of situational awareness [2].

2.2 Trusted Automated eXchange of Indicator Information

TAXII was introduced as a collection of protocols and technical documents which enable exchanging relevant CTI among organizations and different members of industry sectors. To achieve this, TAXII specified a set of data formats and mechanisms for secure transport of threat information which is later used for the detection, prevention, and mitigation of cyberattacks in real-time. From a technological perspective, TAXII is an application protocol and service specification for exchanging CTI over HTTPS. Hence, the TAXII standard conforms to a client-server model and identifies sets of requirements for both the service clients and the respective server. Additionally, TAXII is implemented as a RESTful API providing a set of services and message exchanges. TAXII also defines two primary services to cover the full range of cyber threat information sharing use cases as follows:

- **Collection** : A Collection serves as an interface to a logical repository of CTI objects which are maintained by a TAXII server on behalf of CTI producers and subsequently can be inquired by consumers. Therefore, this service follows the request-response model for communication between TAXII clients and servers.
- **Channel** : Like a collection, a Channel is maintained by a TAXII server and enables CTI producers to publish threat information simultaneously to several consumers. Consequently, this service allows consumers to receive CTI from different producers. To this end, a channel adopts the publish-subscribe communication model for threat information exchange. It is noteworthy that the latest version of the TAXII standard (i.e. v2.1) does not support Channel services yet.

TAXII was initially designed to enable an exchange of CTI data formatted as STIX objects. Thereby, support for transporting STIX objects is mandatory in TAXII standard. However, it is feasible to exchange threat information in other formats using TAXII messages. In other words, STIX and TAXII are fully independent standards which imply that the formats and serializations of STIX objects were not defined considering any specific transport protocols, and TAXII may facilitate sharing of non-STIX data [16].

Authors in [6] believed that the widespread deployment of TAXII may lead to a substantial reduction in operational changes, seamless interoperability with existing sharing policies, and covering a variety of threat sharing paradigms such as peer-to-peer, and source-subscriber.

2.3 Distributed Ledger Technologies

DLTs provide a tamper-proof database where trust is a by-product of the collaboration among a set of computers [38]. Thereby, EPES infrastructure may benefit from DLTs for establishing agreements and secure exchange of CTI data among different components. In DLTs, data records are immutable. This means once a block of data is written to a DLT, it cannot be altered because each data block apart from transaction data and timestamp contains a cryptographic hash of its content as well as the prior block [47]. Consequently, if a block is modified for any arbitrary reasons, its new hash would mismatch the successor data blocks and results in an invalid chain of blocks. Furthermore, within a DLT, several nodes retain the same copy of data records which leads to better availability and greater resiliency against failures compared to traditional databases with centralized architectures. DLTs also offer trust among several unknown nodes in absence of trustworthy third parties. To this end, DLTs employ consensus mechanisms to grant admission for appending a new block in proper order. Due to the aforementioned features, a DLT offers traceability, integrity, availability and reliability as a data storage. Hence, it can play a key role in EPES infrastructure where the huge volume of threat information is generated and exchanged among different components to accelerate cyber-attacks prevention, detection and elimination.

DLTs based on how they control data access rights are categorized as follows: public, private, permissioned and permissionless. In a private permissioned ledger, one organisation manages the read and write access rights to data records thus only certain nodes are authorized to execute various operations on blocks of data. While in a public permissionless ledger, any node can join the ledger network and further carry out different transactions such as read and write on data records. Due to the openness of such DLTs, they apply relatively complex consensus algorithms to guarantee trust among participating nodes which result in an expensive and time-consuming process of data recording. On contrary, the write operation in the private permissioned ledgers is both time and cost-efficient since they utilize consensus mechanisms with lower complexity.

The CTI data produced in an EPES infrastructure cannot be shared with everyone especially components and services outside the EPES premises. Considering the requirements of threat information in EPES and characteristics of different types of DLTs which are described above, private permissioned ledgers such as Quorum [26] and Hyperledger Fabric [1] are seen as appropriate alternatives for storing CTI data within EPES.

3 RELATED WORK

As stated in surveys carried out by SANS [43], the volume of CTI production and consumption continues to grow over the last few years, implying a massive raise in sharing of cyber-security information among disparate players within/across CIs including the power industry. In [34], authors identified the challenges that different sectors encountered while employing various cyber-security information sharing platforms. Kotu et al. [17] explored a few examples of threat information sharing frameworks that were specifically devised for the energy sector.

The cyber-security information sharing platforms has been studied from different perspectives namely data formats, privacy and data authenticity [19]. As described in [14], cyber threat intelligence aims for early detection, prevention and mitigation of cyber-attacks through gathering, sharing and evaluation of threat information. Since cyber threat intelligence is a rather novel field and a substantial percentage of data in this context are either unstructured or encoded in different formats, lately there have been remarkable efforts towards modelling incidents, threat actors and attack patterns which lead to the definition of several ontologies.

These data representations are mostly use-case specific and facilitate the process of sharing cyber threat intelligence [4]. Incident Object Description Exchange Format (IODEF) [8], Open Incident of Compromise (OpenIOC) [22], Cyber Observable eXpression (CybOX^{TM}) [24] , and Structured Threat Information eXpression (STIX) [2] are examples of cyber-security information sharing frameworks[37]. IODEF is an IETF standard and mainly used to exchange incidents among Computer Emergency Response Teams (CERTs) and service providers. The IODEF data models are formatted in XML, thus they can be easily transported across network [5]. OpenIOC is another open framework for sharing CTI data among computers. Organizations utilize this framework which was originally implemented by MANDIANT to disseminate the most recent Indicators of Compromise (IoCs) among themselves. In OpenIOC, the pre-defined set of indicators are formatted using XML and simply can be extended for novel indicators [7]. CybOX is another standardized framework that was designed for modelling and sharing observables including dynamic events and stateful measures. Unlike many cyber intelligence ontologies, CybOX is not tailored for a particular cybersecurity scenario and supports a wide range of use cases from the operational instances of cyber observables to the potential patterns within the cybersecurity realm [24]. As described in 2.1, STIX [2] is community-driven effort that provides a highly flexible and extensible modeling language for CTI data.

To classify numerous cyber-security information frameworks proposed over the last decades, the researchers in [5] introduced a layered taxonomy model. This model was later used by [25] to assess STIX, leading to the conclusion

that STIX can support a full range of concepts within the threat landscape. Additionally, authors in [37] following the extensive survey that was conducted over 22 different cyber-security information sharing frameworks, recognized the STIX standard as a promising solution for modelling and sharing CTI data[37]. Considering the outcomes of the above studies, this work opts for STIX, as an effective modelling language for representing a full range of CTI objects along with the TAXII framework as a transport protocol for secure exchange of threat information.

However, all studies mentioned above collectively inferred the essence of having a common understanding of data models which are utilized for exchanging threat information among different parties as they determine how the data is structured and subsequently need to be processed, but the survey on a large number of CTI providers indicated that threat information is mostly exchanged as plaintext and only a small percentage of CTI sources structure their data into standardized ontologies such as MISP [46] and STIX [34]. Hence, this paper mainly focuses on practical aspects of cyber information sharing platforms specifically the TAXII framework including data integrity, database protection and real-time data exchange which were less target for discussion in state-of-the-art research works.

The cyber-security information sharing platforms have similar security requirements as communication networks namely privacy, integrity and availability[18]. Several works [11, 20, 33] centered on privacy challenges of such platforms. For instance, [9] introduced a cyber-security information sharing network where a combination of format-preserving and homomorphic encryption mechanisms was applied to information represented in STIX data models towards enhancing data privacy. Moreover, Leszczyna et al. [19] combat the privacy issues stemming from sharing CTI data through data sanitisation and anonymisation.

Apart from privacy concerns, in recent years there have been growing efforts to compromise the authenticity of CTI data that is exchanged within/across cyber-security information sharing platforms since these frameworks are less explored from this perspective and are most vulnerable to the attacks targeting data integrity. This work addresses the identified research gap by incorporating DLTs into the TAXII framework, thus benefiting from the immutable audit trail of data in ledgers, leading to tamper-resist CTI data records which are shared with authorized parties. More specifically, the TAXII framework is extended as it persists a hash of CTI data in a dedicated ledger and further uses the hash to validate the authenticity of actual threat information stored in a local database. However, during the last years, some works proposed cyber threat information sharing mechanisms based on blockchains as a resolution to the vital need of data integrity and traceability in such platforms, but they did not mainly go beyond design proposals and lacked practical proofs-of-concept. For instance, authors in [15] introduced a blockchain-based threat intelligence sharing and rating system where smart contracts were devised to rate CTI data, although it did not present an adequate implementation for the proposed solution. Riesco et al. [36] addressed the reluctance of CTI producers to share their thereat information with other interested parties by introducing a cyber threat information marketplace based on Ethereum. Given the fact mining transactions on Ethereum is a relatively long process as opposed to the exchange of threat information that commonly needs to be swift, this work does not offer a practical solution concerning several application scenarios. [31] is another example of the distributed marketplace based on Ethereum where cyber-security services are traded between clients and security professionals.

Furthermore, to effectively counter sophisticated cyberattacks within EPES infrastructure, it is imperative to share CTI data in real-time. To this end, this work complements the TAXII framework with a generalized publish-subscribe middleware, implementing the RabbitMQ messaging platform [21].

4 TAXII FRAMEWORK ENHANCEMENTS - APPLICATION FOR ELECTRICAL POWER AND ENERGY SYSTEM

Motivated by the vital need for sharing CTI data securely and efficiently within EPES infrastructure towards accelerating detection and mitigation of advanced cyberattacks, this section presents a novel approach where the TAXII framework is extended by integrating DLTs and introducing a generalized publish-subscribe middleware. Consequently, the power industry may adopt and benefit from the enhanced TAXII framework as a secure, tamper-resist and scalable cyber-security information sharing platform. Figure 1 depicts the overall architecture of the proposed solution. The TAXII framework expansion gives the ability to CTI producer and consumer to confirm that threat information has not been modified. Along with enhancing data integrity, the generalized publish-subscribe middleware creates an opportunity for real-time and scalable sharing of CTI data between different components and services. As shown in Figure 1, this framework comprises three main components: TAXII client, Medallion node (i.e. TAXII server) and publish/subscribe middleware. The implementation details of each component are thoroughly described in the following sections. It is worth mentioning our prototype is built on top of the minimal implementation of TAXII 2.1 standard which is publicly available at OASIS TC open repository [27, 28].

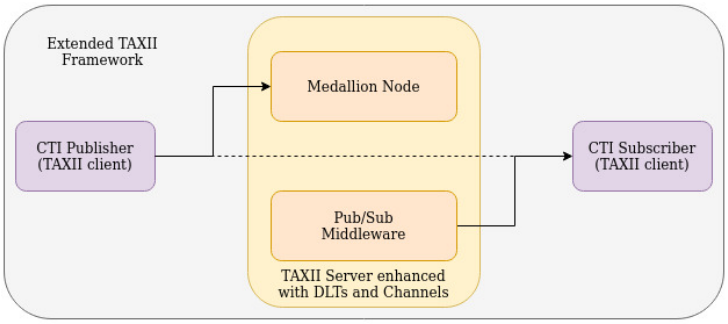

Fig. 1. Architecture of the enhanced TAXII framework which comprises a generalized pub/sub middleware in addition to TAXII client and server

4.1 TAXII Client

Many components within the context of EPES infrastructures produce a substantial amount of CTI data such as anomaly reports and threat logs which need to be shared securely and efficiently with other services and components. To achieve this goal, the EPES components, and services regardless of their roles which might be either CTI producer, consumer, or both, must support the TAXII client that is a reference implementation of TAXII 2.1 client [27]. This prototype enables message exchange between clients through a medallion node which maintains a repository of CTI objects provided by CTI producers and subsequently replies to consumer queries. The TAXII 2.1 server implementation and related functionalities are extensively discussed in the following section.

4.2 TAXII Server

The key goal of the TAXII server is to disseminate threat information between CTI providers and consumers. The TAXII server in our prototype is implemented on basis of medallion which was mainly developed as a minimal implementation

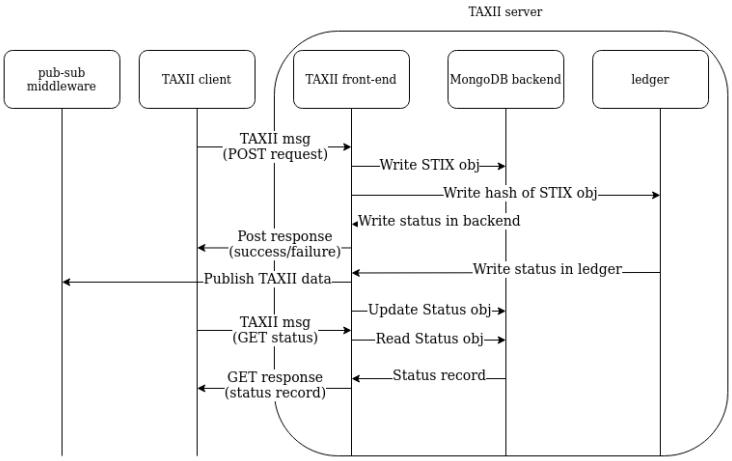

Fig. 2. Sequence diagram followed by TAXII server for processing a POST request. This procedure, apart from recording STIX objects in the MongoDB backend, undertakes writing hashes to a dedicated ledger

of TAXII 2.1 server[28]. Medallion is a simple front-end REST server in python, making the endpoints specified in TAXII 2.1 protocol accessible for TAXII clients. To this end, it uses Flask [32] which is a micro web framework. Medallion, in addition, supports different back-end "plugins". These plugins were devised to persist the TAXII data and metadata. Thereby, the TAXII server enables clients to either write information to its backend or read data from it through different interfaces so-called TAXII Collections. In the described procedure, the TAXII server and clients abide by the request-response communication model. The medallion node in this prototype is configured with the MongoDB backend which runs as a separate server.

As stated above, the TAXII server stores CTI data received from various clients in its backend. This mechanism, however, does not guarantee the authenticity of STIX objects maintained by the TAXII server. To address this deficiency, our TAXII server uses private permissioned DLTs namely Quorum and Hyperledger Fabric as immutable data storages to ensure the integrity of CTI data records in its backend. To persist CTI data, the TAXII server performs the following steps which are also presented in Figure 2.

(1) The TAXII server's front-end upon reception of a POST request from a client, writes the TAXII data and metadata, encompassing one or more STIX objects, in the MongoDB backend.
(2) At the same time, it triggers writing a hash of every STIX object carried by the TAXII message in a dedicated ledger. To make this process feasible, ledger-specific adapters are added to the official TAXII server. These adapters allow the TAXII server to execute transactions on different types of the ledger. More precisely, ledger-specific adapters provide interfaces to the pre-defined smart contracts on respective ledgers. Consequently, the TAXII server can simply invoke functions on target ledgers. For now, the TAXII server only includes adapters for QUORUM [3] and Hyperledger Fabric [41] ledgers, however, support for other ledger types can be easily added. As mentioned above, only the hash of STIX objects is written in ledgers. The rationale behind this design decision is the STIX objects generated by EPES components might explicitly or implicitly contain personal data thus storing them in ledgers may lead to violation of the GDPR right to be forgotten due to the immutable nature of

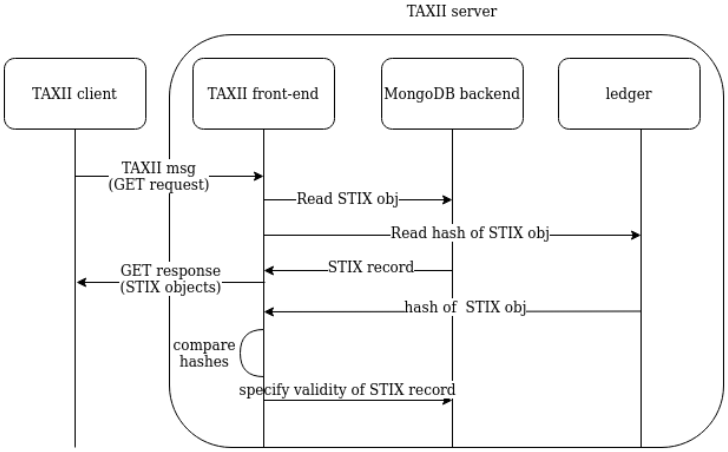

Fig. 3. Sequence diagram followed by TAXII server for processing a GET request. This mechanism verifies the authenticity of a STIX record in the backend by comparing its hash with the related hash stored in the ledger

DLTs. To generate a hash of STIX objects, the TAXII server applies the MD5 hash function which is a one-way function and computationally very difficult to invert.

(3) After completion of the write operation on the database, the MongoDB backend sends back the operation status to the TAXII front-end.
(4) The TAXII front-end creates a status resource per the request status provided by the backend and report it back to the client. The status resource pertains to the state of all STIX objects in the POST request and specifies whether the write operation related to each object is still pending, completed or failed. The TAXII front-end then persists the status resource in the MongoDB backend.
(5) The TAXII 2.1 does not define the methods and data formats for the publish-subscribe communication paradigm despite the fact it has been promised in the standard specification. Therefore, this prototype develops a separate module (i.e. pub/sub middleware), enabling real-time applications and event-driven architecture. Thereby, the TAXII server right after sending the POST response passes the TAXII data to the pub/sub middleware where STIX objects are forwarded to pre-defined channels.
(6) After finalizing publishing data on the deployed ledger, the status resource associated with the STIX object is updated accordingly. More accurately, the status is set to either completed if the write operation in the ledger was successful or failure in case of an unsuccessful attempt to write the data.
(7) The TAXII client can later inquiry about the status of each STIX object using a separate TAXII message (i.e. GET status request). In this manner, the client can ensure the TAXII data and related metadata is successfully written in both MongoDB backend and the ledger.

In EPES infrastructure, components and services mostly receive the published CTI data through the channels dedicated to different types of STIX object. However, some components may follow the request-response communication paradigm for obtaining CTI data. To this end, they send a TAXII message carrying a GET request for a certain STIX object directly to the TAXII server. To handle GET requests, the TAXII server takes the following steps as illustrated in Figure 3 .

(1) Upon reception of a GET request which is encapsulated in a TAXII message, the TAXII front-end retrieves the queried STIX objects from the MongoDB backend.
(2) Simultaneously, the TAXII front-end triggers reading of the hash of the related STIX object from the ledger through the related ledger-specific adapter.
(3) When the front-end receives the hash of the object from the ledger, it compares that with the hash of the actual STIX object provided by the backend. The identical hashes imply that the STIX record on the backend is valid while the discrepancy in the hashes indicates some modification in the data record. In the latter case, the front-end notifies the backend about the discrepancy and subsequently the backend labels the STIX record as an invalid entry. It is noteworthy that the front-end rejects any GET request aimed at the invalid STIX object. Thereby, this procedure can detect any malicious behaviours aimed at modifying STIX records in the MongoDB backend which results in a higher degree of data integrity.

4.2.1 Authentication on TAXII Server. The official TAXII server supports HTTP Basic authorization, however, it provides an opportunity for developers to employ more advanced authorization and authentication mechanisms. [28] In our prototype, the TAXII server uses Keycloak, a central Authorization, Authentication, Accounting (AAA) OAuth2 service [40]. To access the server for read or write operation, the client must have access to an account with access to the "taxii-read" and "taxii-write" scopes respectively. The TAXII server can be configured on deployment to use the correct Keycloak instance and details namely URL, client id, client secret and realm. When making an http request to the server, the client must provide authorization credentials in one of two ways:

(a) Using a valid JSON Web Token obtained independently from the Keycloak instance, that provides access to the correct scopes as mentioned above. The token is sent in the 'Authorization' header with the prefix 'Token ' and validated in the TAXII server.
(b) Using Basic authentication, as per the TAXII protocol. In this case, the TAXII server uses the credentials to obtain authorization from Keycloak on behalf of the client, using the OAuth2 password flow. Basic Authorization is defined in RFC 7617 [35].

4.2.2 Ledger-specific Adapter. As described in 4.2, the TAXII server communicates with ledger nodes via ledger-specific adapters. These adapters provide interfaces toward different ledger types. To this end, each adapter first setups a connection between the TAXII server and the corresponding ledger type. After that, for the write operation, it triggers a transaction by invoking the pre-defined function from the configured smart contracts on the ledger. Subsequently, every function calls that carries the hash of the STIX object as an input, will be persisted in the ledger. For the read operation, the adapter uses the transaction IDs correspond to the write operation to retrieve the STIX object hashes from the ledger that will be further examined against the hashes of the STIX records in the TAXII backend.

4.3 Publish-Subscribe Middleware

In general, the use of a publish-subscribe mechanism facilitates the real-time propagation of CTI data regarding indicators, attacks, or mitigation actions through the various components. Such a mechanism is proposed by TAXII standard, but not yet part of the latest version of the protocol, and thus not supported by Medallion. Therefore, the official TAXII 2.1 server implementation has been extended to develop the channels missing specification. Components that produce CTI data and post it on the TAXII server can add metadata specifying channels that it will be forwarded to, and components subscribed to these channels will receive them.

4.3.1 Channel specification metadata. The TAXII server will forward STIX bundles containing a marking definition object with the following field:

```
"definition":{ "statement": "pubsub:X" }
```

The bundle will be forwarded to channel X, where X is one of the available channels, which are by default categorized to *EVENTS*, *ATTACKS*, *MITIGATIONS* and *UNSPECIFIED* channels and configured on deployment. Created objects outside of bundles, or bundles not containing this marking definition, will not be forwarded to any channels.

The pub/sub mechanism is generically specified using a concrete interface specification. The current implementation supports the RabbitMQ messaging platform, though the provided middleware supports dependency injection and can be, thus, extended to use another messaging platform, such as Kafka.

4.3.2 RabbitMQ architecture. On start-up, the TAXII server (the publisher) creates one RabbitMQ exchange of type 'fanout' for each supported channel if it does not already exist. Messages are published on their corresponding exchange, based on the marking definition as specified above. Subscribers then create temporary queues for each topic they subscribe to, bind them to the corresponding exchange, and start consuming.

5 EXPERIMENTAL RESULTS AND EVALUATION

To measure overhead incurred by the proposed solution while publishing TAXII data and metadata or retrieving relevant threat information, a similar system structure as illustrated in Figure 1 is used where the medallion node with the MongoDB backend is configured to integrate two distinct Quorum ledgers. Additionally, the smart contract that defines a function aimed at publishing data is deployed on both ledgers. The process of smart contract deployment and all relevant transactions on Quorum nodes are feeless due to the private nature of Quorum ledgers. The official TAXII framework is used as a based line for evaluating the applicability of our design considering performance indicators such as cost and delay. All experiments were carried out on a laptop computer with 16GB of memory and 1.6GHz Intel i5-8365U CPU and respective results are listed in Table 1.

Table 1. Write and read operations in TAXII framework

Action	**Official TAXII**	**Extended TAXII**
	Operation Timing (s)	
Publishing CTI data	0.025	1.29
Retrieving CTI data	0.029	0.18

In Table 1, the timings for publishing CTI data on both our framework and the official implementation is presented. The measurements were considered as follows: first, a TAXII client, acting as a CTI publisher, sends a POST request carrying two STIX objects to the medallion node. In our design, the time of publishing CTI data is 1.29s while the official framework stores the identical data in 0.025 second. This time difference arises from the fact that our TAXII server in addition to persisting a STIX object in the backend that is a standard way of handling a POST request, stores its hash in Quorum ledgers. As results demonstrate, the overhead of write operation in Quorum ledgers is higher due to the interaction with the DLT; the proposed design trades timing for assuring the authenticity of data and eliminating malicious behavior targeting the integrity of CTI data.

After finalizing the publishing of CTI data, the TAXII client, dispatches two HTTP GET requests against TAXII messages to validate and retrieve the previously published STIX objects. As presented in Table 1, the time required for

reading this data from our TAXII server and the official medallion node are 0.18 and 0.029 seconds respectively. The higher delay of read operation in our framework stems from the need for additional transactions aiming towards the retrieval of the hash of the requested objects.

6 DISCUSSION AND FUTURE WORK

Following the use case and initial evaluation results presented in section 5, the proposed set of extensions to the TAXII protocol and service specification was reference implemented, deployed and evaluated in a lab environment, to address CTI data exchange needs of EPES infrastructures at both local (single- or multi-site infrastructures) and regional/national/international levels. Although the original and primary target of this evaluation was related to the modelling and sharing CTI data enablement, the scope of the relevant application (i.e. exchange of CI-related CTI), called for extra requirements particularly as to the data integrity and non-repudiation, to and, at a second level, to its integration into a mission-critical and event-based architecture.

The proposed solution satisfies the posed requirements, making use of state-of-the-art open source technologies and frameworks, effectively and by design guaranteeing data integrity using DLTs, preventing unauthorized access through OAuth2.0 integration into the TAXII framework and enabling real-time and controlled streaming of security-related notifications to the designated architectural components through the adopted publish-subscribe middleware. The interaction with the DLT introduces delays in the standard TAXII operations. Since the proposed framework trades processing time for security and data non-repudiation assurance, it is particularly valuable when priority is put on the latter, but would not be valid in cases of delay-intolerant operations.

As the next steps, the proposed framework will be extended with support for more messaging frameworks (e.g. Kafka) as they would implement the Channels approach of TAXII. Moreover, the current implementation will be further improved with the integration of the Interledger module that combines the strengths of different ledgers while combating their shortcomings. The Interledger integration allows the TAXII framework to use different ledger technologies for varying use cases. For instance, the ledgers aimed for persisting full CTI data records are preferably chosen from private ledgers to achieve better throughput and lower cost whereas public ledgers may be used for storing a hash of threat information offering a higher level of trust.

7 CONCLUSION

In this paper, a set of extensions to the TAXII protocol and service specification for exchanging CTI data has been presented. The extensions grant the associated TAXII services with data non-repudiation and real-time operation characteristics. The extended TAXII service specification was tested and positively validated in lab environment, reasonably trading read/write performance for increased assurances towards data repudiation and event-driven architectural patterns enablement.

ACKNOWLEDGMENTS

This work was partially funded by the H2020 PHOENIX project, contract no. 832989, within the H2020 Framework Program of the European Commission.

REFERENCES

[1] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. 2018. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings*

of the thirteenth EuroSys conference. 1–15.

[2] Sean Barnum. 2012. Standardizing cyber threat intelligence information with the structured threat information expression (stix). *Mitre Corporation* 11 (2012), 1–22.

[3] Blockgeeks. 2020. *What Is Quorum Blockchain? A Platform for The Enterprise.* https://blockgeeks.com/guides/quorum-a-blockchain-platform-for-the-enterprise/#View_of_Party_B

[4] Fabian Böhm, Florian Menges, and Günther Pernul. 2018. Graph-based visual analytics for cyber threat intelligence. *Cybersecurity* 1, 1 (2018), 1–19.

[5] Eric W Burger, Michael D Goodman, Panos Kampanakis, and Kevin A Zhu. 2014. Taxonomy model for cyber threat intelligence information exchange technologies. In *Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security*. 51–60.

[6] Julie Connolly, Mark Davidson, and Charles Schmidt. 2014. The trusted automated exchange of indicator information (taxii). *The MITRE Corporation* (2014), 1–20.

[7] Cyware. 2019. *What is Open Indicators of Compromise (OpenIOC) Framework?* https://cyware.com/educational-guides/cyber-threat-intelligence/what-is-open-indicators-of-compromise-openioc-framework-ed9d

[8] R Danyliw. 2016. The Incident Object Description Exchange Format Version 2. *RFC Editor* (2016).

[9] José M de Fuentes, Lorena González-Manzano, Juan Tapiador, and Pedro Peris-Lopez. 2017. PRACIS: Privacy-preserving and aggregatable cybersecurity information sharing. *computers & security* 69 (2017), 127–141.

[10] US DHS. 2019. Critical infrastructure sectors. *US Department of Homeland Security.[Online].[Accessed 2015-2019]* (2019).

[11] Larry A Dunning and Ray Kresman. 2012. Privacy preserving data sharing with anonymous ID assignment. *IEEE Transactions on Information Forensics and Security* 8, 2 (2012), 402–413.

[12] A EC. 2010. Digital Agenda for Europe, com (2010) 245 final.

[13] Ivo Friedberg, Florian Skopik, Giuseppe Settanni, and Roman Fiedler. 2015. Combating advanced persistent threats: From network event correlation to incident detection. *Computers & Security* 48 (2015), 35–57.

[14] Mari Grønberg. 2019. *An Ontology for Cyber Threat Intelligence.* Master's thesis. University of Oslo.

[15] Shen He, Jun Fu, Wen Jiang, Yexia Cheng, Jiake Chen, and Zhihui Guo. 2020. BloTISRT: Blockchain-based Threat Intelligence Sharing and Rating Technology. In *Proceedings of the 2020 International Conference on Cyberspace Innovation of Advanced Technologies*. 524–534.

[16] Bret Jordan and Drew Varner. 2020. TAXIITM Version 2.1,Committee Specification 01. *OASIS Cyber Threat Intelligence (CTI) TC* (2020).

[17] Lindah Kotut and Luay A Wahsheh. 2016. Survey of cyber security challenges and solutions in smart grids. In *2016 cybersecurity symposium (CYBERSEC)*. IEEE, 32–37.

[18] A Lee and T Brewer. 2014. Guidelines for Smart Grid Cyber security, Volume 1, Smart Grid Cybersecurity Strategy, Architecture, and High Level Requirements.

[19] Rafał Leszczyna, Maciej Łosiński, and Robert Małkowski. 2015. Security information sharing for the polish power system. In *2015 Modern Electric Power Systems (MEPS)*. IEEE, 1–6.

[20] Xuefeng Liu, Yuqing Zhang, Boyang Wang, and Jingbo Yan. 2012. Mona: Secure multi-owner data sharing for dynamic groups in the cloud. *IEEE transactions on parallel and distributed systems* 24, 6 (2012), 1182–1191.

[21] Rabbit Technologies Ltd. 2007. *RabbitMQ.* https://www.rabbitmq.com/

[22] Mandiant. 2010. *OpenIOC.* http://www.openioc.org/

[23] Mirco Marchetti, Fabio Pierazzi, Michele Colajanni, and Alessandro Guido. 2016. Analysis of high volumes of network traffic for advanced persistent threat detection. *Computer Networks* 109 (2016), 127–141.

[24] MITRE. 2011. *Cyber Observable eXpression.* https://cybox.mitre.org/about/

[25] MITRE. 2018. *Sharing threat intelligence just got a lot easier.* www.oasis-open.github.io

[26] Patrick Mylund Nielsen and David Voell. 2016. *Quorum: an Ethereum-forked variant Blockchain.* https://github.com/jpmorganchase/quorum-docs/blob/master/Quorum%20Whitepaper%20v0.1.pdf

[27] OASIS. 2017. *cti-taxii-client.* https://github.com/oasis-open/cti-taxii-client

[28] OASIS. 2017. *cti-taxii-server.* https://github.com/oasis-open/cti-taxii-server

[29] OASIS. 2017. *STIX Version 2.1 Committee Specification 02.* https://oasis-open.github.io/cti-documentation/resources#stix-21-specification

[30] Ijeoma Onyeji, Morgan Bazilian, and Chris Bronk. 2014. Cyber security and critical energy infrastructure. *The Electricity Journal* 27, 2 (2014), 52–60.

[31] Polyswarm. 2017. *Polyswarm decentralized threat detection market- place.* https://polyswarm.io

[32] Pallets project. 2010. *Flask-web development, one drop at a time.*

[33] Abhinav Raj, R Arunprasath, and S Vigneshwari. 2016. Efficient mechanism for sharing private data in a secured manner. In *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*. IEEE, 1–4.

[34] Konstantinos Rantos, Arnolnt Spyros, Alexandros Papanikolaou, Antonios Kritsas, Christos Ilioudis, and Vasilios Katos. 2020. Interoperability challenges in the cybersecurity information sharing ecosystem. *Computers* 9, 1 (2020), 18.

[35] Julian Reschke. 2015. The'basic'http authentication scheme. *Work in Progress, draft-ietf-httpauth-basicauth-update-07* (2015).

[36] R Riesco, X Larriva-Novo, and VA Villagra. 2020. Cybersecurity threat intelligence knowledge exchange based on blockchain. *Telecommunication Systems* (2020), 259–288.

[37] Clemens Sauerwein, Christian Sillaber, Andrea Mussmann, and Ruth Breu. 2017. Threat intelligence sharing platforms: An exploratory study of software vendors and research perspectives. (2017).

[38] Vasilios A Siris, Pekka Nikander, Spyros Voulgaris, Nikos Fotiou, Dmitrij Lagutin, and George C Polyzos. 2019. Interledger approaches. *IEEE Access* 7 (2019), 89948–89966.

[39] Chih-Che Sun, Adam Hahn, and Chen-Ching Liu. 2018. Cyber security of a power grid: State-of-the-art. *International Journal of Electrical Power & Energy Systems* 99 (2018), 45–56.

[40] Synelixis. 2021. *Open Source Identity and Access Management For Modern Applications and Services.* https://www.keycloak.org/

[41] P. Thummavet. 2019. *Demystifying Hyperledger Fabric (1/3): Fabric Architecture.* https://medium.com/coinmonks/demystifying-hyperledger-fabric-1-3-fabric-architecture-a2fdb587f6cb

[42] Deepak Tosh, Shamik Sengupta, Charles A Kamhoua, and Kevin A Kwiat. 2018. Establishing evolutionary game models for cyber security information exchange (cybex). *J. Comput. System Sci.* 98 (2018), 27–52.

[43] Wiem Tounsi and Helmi Rais. 2018. A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Computers & security* 72 (2018), 212–233.

[44] Iman Vakilinia and Shamik Sengupta. 2017. A coalitional game theory approach for cybersecurity information sharing. In *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*. IEEE, 237–242.

[45] Erik Van der Vleuten and Vincent Lagendijk. 2010. Transnational infrastructure vulnerability: The historical shaping of the 2006 European "Blackout". *Energy Policy* 38, 4 (2010), 2042–2052.

[46] 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 on Workshop on Information Sharing and Collaborative Security*. 49–56.

[47] Xu Wang, Xuan Zha, Wei Ni, Ren Ping Liu, Y. Jay Guo, Xinxin Niu, and Kangfeng Zheng. 2019. Survey on blockchain for Internet of Things. *Computer Communications* 136 (2019), 10–29. https://doi.org/10.1016/j.comcom.2019.01.006