Privacy-Preserving Correlation of Cross-Organizational Cyber Threat Intelligence with Private Graph Intersections

Davy Preuveneers and Wouter Joosen

[a]*imec-DistriNet, KU Leuven, Celestijnenlaan 200A, Leuven, B-3001, Belgium*

Abstract

Sharing cyber threat intelligence is important because it allows organizations to stay ahead of new and emerging threats, prevent downtime and improve their overall security posture. Information about known vulnerabilities and post-mortem analyses of successful attacks are instrumental to make tactical decisions and implement adequate countermeasures. However, organizations are hesitant or cautious to share their locally collected cyber threat intelligence with third parties because of possible damage to the organization's reputation, legal or liability concerns, or the risk that the information is used against them.

In order to promote a collaborative cybersecurity environment that accommodates the varying confidentiality requirements of both threat intelligence producers and consumers, we introduce and assess a viable solution for preserving privacy while sharing and analyzing sensitive or confidential data. This solution is designed to work seamlessly with modern cyber threat intelligence platforms. Furthermore, we examine the security implications and computational impact associated with these techniques, enabling the analysis of correlations between threat events in a manner that respects confidentiality and extends across multiple organizations involved in information sharing.

Keywords: threat intelligence sharing, security, privacy, polyglot persistence and analysis, private graph intersection

1. Introduction

Threat intelligence platforms (TIPs) have gained widespread adoption among organizations, enterprises, and CERT communities for analyzing security incidents and gaining valuable insights to mitigate cyber attacks. With the increasing number and complexity of security threats and attack campaigns, organizations can no longer rely solely on their own resources to effectively and proactively defend against attacks. To enhance collaborative situational awareness, various data exchange formats like STIX [1], TAXII [2], and CyBOX [3] have been proposed and integrated into TIPs to facilitate the sharing of indicators of compromise (IoCs). This enables organizations to easily incorporate diverse intelligence feeds, such as CIRCLE OSINT[1], Botvrij.eu[2], or the Feodo IP Blocklist[3], to enrich the threat intelligence gathered from their own systems and networks. While collecting and sharing security incident data are crucial TIP functionalities, the key aspect lies in efficiently sifting through vast intelligence feeds [4] to filter and extract actionable information that is relevant for safeguarding the organization, its systems, and networks.

Despite the availability of technical means to facilitate information sharing, organizations exhibit reluctance when it comes to sharing their own threat intelligence. This hesitation stems from concerns about reputational damage when customers become aware of a breach and the risk of publicly exposing sensitive or private information. The General Data Protection Regulation (GDPR) imposes strict restrictions on the publication of personally identifiable information (PII) and carries substantial penalties for non-compliance. An example of the GDPR's impact is evident in WHOIS [5], a service commonly used by security analysts to obtain information about domain names or websites. Due to GDPR compliance, WHOIS now redacts or anonymizes certain registrant details [6] such as names, addresses, email addresses, and phone numbers. Consequently, when a domain name is employed for malicious purposes,

[1] https://www.circl.lu/doc/misp/feed-osint/
[2] https://www.botvrij.eu/data/feed-osint/
[3] https://feodotracker.abuse.ch/downloads/ipblocklist.csv

these attributes are no longer accessible to security analysts, hindering their ability to trace threats and identify the perpetrators behind attack campaigns. Even an IP address, often utilized as an indicator of compromise (IoC), falls under the scope of personal data as per the GDPR if it can be linked to an identified or identifiable 'natural' person[4].

Enabling the sharing of threat intelligence is crucial for effective attack prevention. However, concerns surrounding confidentiality and privacy often hinder voluntary reporting efforts. To address this challenge, we propose a practical solution for threat intelligence platforms (TIPs) that enables security analysts to strike a better balance between security and privacy when dealing with sensitive business information, private data, and personally identifiable information. Our solution, which builds upon our prior research [7, 8, 9], implements a polyglot framework that integrates various privacy-enhancing techniques for storing, processing, and sharing threat intelligence. This approach caters to the diverse security and privacy requirements of both threat intelligence producers and consumers. While the term 'polyglot persistence' [10] typically refers to using different technologies for managing varied data storage needs, our polyglot solution extends this concept to encompass multi-faceted processing and sharing of information. Specifically, threat intelligence is made available in different forms through selective suppression, encoding, hashing, and encryption—both individually and in combination—based on the information's sensitivity and the security analytics requirements of authorized recipients. Moreover, we explore the security implications and computational overhead of these techniques within the MISP [11] cyber threat intelligence platform to analyze correlations between threat events from different organizations while preserving privacy. These techniques leverage Private Set Intersection (PSI) [12], a privacy-preserving cryptographic technique that enables two parties to compare their data sets and compute intersections without exposing raw data to the other party.

This study builds upon and extends our earlier award-winning research [13]. In this research, we distinguish it from our prior work in several ways. First, we enhance and expand the discussion of related work, discussing additional research as well as providing more comprehensive insights. Second, we present a more detailed explanation of our framework and its underlying technical building blocks. Third, we conduct additional experiments using new datasets that contain a significantly larger number of threat events and attributes, enriching the evaluation process. Lastly, we present a novel baseline experiment for comparison, which incorporates a trusted third party responsible for computing the set intersection. This third party facilitates threat intelligence producers and consumers in discovering the information they have in common with reduced computational overhead. However, this approach operates under a distinct threat model, as both parties must assume that the third party will not engage in malicious behavior or collude with either of them.

1.1. Contributions

The key contributions of this work can be summarized as follows:

1. Our framework introduces a polyglot persistence and analysis approach for threat intelligence, accommodating the distinct requirements of both intelligence feed producers and consumers. By incorporating a polyglot approach, it enables threat intelligence producers to employ more refined sharing mechanisms. This approach allows them to contribute to the security community in a diversified manner without compromising the sensitivity or confidentiality of the gathered threat intelligence.
2. We put forward a novel methodology that enables the analysis of correlations between threat events while preserving privacy, extending across multiple organizations engaged in information sharing. This methodology leverages private set intersections and Bloom filters to selectively share threat intelligence information with consumers. It ensures that the shared data is beneficial to the consumers without requiring the provider to have knowledge of the specific information the consumer can access.
3. We conduct an in-depth examination of the security implications and computational overhead associated with diverse privacy-enhancing techniques when integrated into a modern threat intelligence platform. Additionally, we evaluate our approach by contrasting it with an equivalent deployment relying on a trusted third party. This third party computes the intersections using the data from both parties, while ensuring that the disparate data remains undisclosed to each other.

[4] https://gdpr-info.eu/issues/personal-data/

1.2. Overview

The paper is structured as follows: In Section 2, we delve into relevant related work. Section 3 presents the design and implementation details of our polyglot persistence and sharing solution. Section 4 elaborates on the privacy-preserving correlation of threat intelligence. The evaluation of security and privacy impact, as well as the computational complexity, is covered in Section 5. Lastly, in Section 6, we conclude the work by summarizing the key insights and proposing potential avenues for future research.

2. Related work

In this section, we will explore relevant literature on threat intelligence platforms, focusing on their added value, key functionalities such as correlation analysis, and the secure sharing and processing of threat information while prioritizing privacy.

2.1. Threat intelligence sharing and supporting platforms

A recent survey conducted by Zibak et al. [14] aimed to investigate the factors influencing the effectiveness of threat intelligence platforms. The study involved analyzing responses from 152 security professionals in order to gain a deeper understanding of the key factors for success. Their empirical evaluation highlighted that the quality of the information and the perceived trust in the platform are crucial factors for achieving success. Similarly, Li et al. [4] conducted comparative research on the significance of cyber threat intelligence. The authors also verify that numerous public and commercial sources disseminate threat intelligence data. However, it remains uncertain how much these sources truly enhance the defense of systems against future attacks. Consequently, the authors establish a collection of metrics to describe the wide array of sources for threat intelligence and evaluate their extent of coverage and accuracy. Their findings reveal substantial variations in the types of data captured, and there is no proof that larger threat intelligence feeds offer more significant information. Furthermore, they observe a low level of overlap between different data sources. In a more recent study conducted by Bouwman et al. [15], a large information sharing community known as the COVID-19 Cyber Threat Coalition was investigated. With over 4000 members, the study aimed to gain insights into whether collaboration at such a scale resulted in improved coverage and whether the availability of threat data for free enhanced the capability of defenders to take appropriate actions. The findings indicated that the community indeed bolstered the ability of network defenders to respond by disseminating their threat data through a freely accessible blocklist. However, over time, the Cyber Threat Coalition encountered challenges in maintaining focus and struggled to scale up its quality assurance processes, causing them to fall behind established defense mechanisms.

Gascon et al. [16] introduced Mantis, a threat intelligence platform designed to retrieve information and correlate threat data from various threat intelligence standards. The primary objective of Mantis is to assist security analysts in identifying similar attack patterns across seemingly unrelated attack campaigns. To achieve this, Mantis employs a type-agnostic similarity algorithm based on attribute graphs. Through an extensive evaluation involving more than 14000 CyBOX objects, the platform demonstrated its capability to retrieve pertinent threat reports with an impressive mean average precision of 80%. This level of precision is achieved even when provided with just a single object from an incident, such as a file or an HTTP request. In a similar vein, Thom et al. [17] conducted a study on the effectiveness of correlating cyber threat intelligence data, but with a focus on global honeypots that simulate real Internet-facing services. Their aim was to analyze attack and traffic patterns to gain insights into the tactics employed by adversaries. In their study, a total of six multi-service honeypots were strategically deployed across various locations worldwide to collect and categorize network traffic over an extended period spanning from March to December 2020. Their analysis encompassed a wide range of characteristics, including source and destination IP addresses, port numbers, usernames and passwords used, executed commands, and downloaded file types. The authors concluded that their approach of gathering data from geographically distinct zones over an extended duration facilitated a better understanding of attacker intent and methodologies. It also aided in the development of effective strategies to identify malicious behavior and sources of attacks, ultimately serving as a valuable source of cyber threat intelligence.

Gonzalez Granadillo et al. [18] introduced Enriched Threat Intelligence Platform (ETIP), an advanced threat intelligence platform that enhances existing TIPs (Threat Intelligence Platforms) through extended capabilities in import, quality assessment processes, visualization, and information sharing. ETIP effectively combines OSINT

(Open Source Intelligence) data, external data sources, and an organization's internal IT infrastructure to provide enriched threat intelligence. ETIP acquires structured cyber threat data from diverse sources and conducts correlation analysis using both static and dynamic data from external sources and the monitored infrastructure. This enables the generation of a threat score through heuristic-based analysis, enriching the information obtained from OSINT and other sources. The resulting comprehensive output is then shared with external entities, including SIEM systems, for further in-depth analysis and dissemination among trusted organizations. Martins et al. [19] confirm that TIPs offer numerous advantages that allow organizations to efficiently initiate crucial processes such as collecting, analyzing, and sharing threat-related information. Nevertheless, current TIPs face certain limitations that hinder their widespread adoption. The authors designed and evaluated an Automated Event Classification and Correlation Platform (AECCP) to address some of these limitations and enhance the functionality of TIPs. AECCP focuses on improving the quality of threat intelligence by classifying it according to a unified taxonomy, filtering out low-value information, enriching it with valuable data from open-source intelligence sources, and aggregating related information pertaining to the same threat. The effectiveness of AECCP was validated and evaluated using three datasets of events, and it was compared with two other platforms. The results demonstrate that AECCP is capable of automatically generating high-quality threat intelligence, thereby assisting security analysts in analyzing security incidents more efficiently within a shorter timeframe.

Sun et al. [20] conducted a targeted investigation on cyber threat intelligence (CTI) mining to promote a more proactive cybersecurity defense strategy. In their survey, they observed that many organizations primarily concentrate on integrating threat data feeds into their existing network and firewall systems, intrusion prevention systems, and Security Information and Event Management systems (SIEMs). However, they often fail to leverage the valuable insights that such new intelligence can offer. To address this, the authors propose CTI mining as a promising opportunity. This process involves uncovering, processing, and analyzing essential information about cyber threats from multiple data sources. In their study, they provide a comprehensive overview of the most significant works on CTI and present a taxonomy to classify various cybersecurity-related entities and events. This taxonomy encompasses cyber attack tactics, techniques, and procedures, profiles of hackers, indicators of compromise, vulnerability exploits, malware implementations, and threat hunting strategies.

2.2. Analysis of private or confidential threat intelligence

Weathersby [21] conducted a study on the prevalence of personal identifiable information (PII). Specifically, the author examined public malware sandbox samples to investigate the implications of PII for privacy and the sharing of threat intelligence. Through exploratory observation analysis of 1012 random samples of non-malicious PDF files uploaded to online malware scanners, it was found that 72% of the samples contained multiple PII indicators. The majority of the samples analyzed did not contain sensitive information beyond the author's name. However, a small percentage of the samples did contain potentially sensitive financial data, such as credit card numbers, as well as identifying information like phone numbers and IP addresses.

Trocoso-Pastoriza et al. [22] propose a secure framework that facilitates distributed and privacy-preserving sharing of threat intelligence. They highlight the challenge of securing an ever-increasing volume of data, leading to bottlenecks in threat intelligence sharing. The authors' solution is built upon the MISP platform and offers participant organizations the means to leverage their sensitive cyber threat intelligence effectively. The framework provides scalable software and orchestrates collaborations that yield statistically significant and valuable insights. These insights, in turn, support and enhance the efficacy and reliability of implemented cyber defense processes. The approach leverages federated learning and cryptographic techniques based on multiparty homomorphic encryption [23]. The objective is to enable efficient and scalable computation of aggregate statistics and machine learning models on encrypted distributed data. The framework allows for the secure release of either the model itself or only the predictions generated by the model. The authors validate their solution through three representative scenarios for CTI sharing: calculating statistics on the collective dataset (e.g., global number of intrusion events per type of malware), automatic prediction of threat levels using MISP events, and training and detection of DDoS attacks.

In the realm of privacy-preserving threat intelligence, van de Kamp et al. [24] conducted research on cryptographic schemes for the secure sharing of Indicators of Compromise (IoCs) and the reporting of sightings. They employ a cryptographic approach to conceal the specific details of an indicator of compromise, allowing it to be shared with other parties. These parties can still detect intrusions using these cryptographic indicators. Additionally, they apply another cryptographic construction that allows parties to report the number of sightings they have observed to a central

party. This central party can aggregate the messages from multiple parties to determine the total number of sightings for each indicator, without gaining knowledge of the individual party's specific number of sightings.

Dara et al. [25] bring attention to the fact that various cloud-based services, including Google Safe Browsing, offer threat intelligence related to advanced persistent threats (APTs). However, accessing such services typically requires users to disclose their browsing history and files to determine whether their machines have been compromised, which raises concerns about privacy. The authors identify two key advancements necessary for designing privacy-preserving threat intelligence services: (i) privately retrieving elements using keyword(s), and (ii) privately retrieving matching documents. To address these challenges, they employ homomorphic encryption (HE) and private information retrieval (PIR) techniques to safeguard the privacy of users querying public threat intelligence services and databases. Similarly, van Rijswijk-Deij et al. [26] developed DNSBloom, a system that uses Bloom filters as a privacy-enhancing technology to store DNS requests in the context of privacy-conscious threat intelligence. Bloom filters function as a probabilistic set, where a membership test provides a high probability of membership (with a small chance of false positives) or confirms non-membership. By not storing original information and aggregating queries from multiple users within specific time intervals, DNSBloom ensures robust privacy protection. At the same time, security professionals can confidently examine whether specific DNS queries associated with malicious activities have occurred with a high level of certainty. Freudiger et al. [27] presented the Sharing is Caring (SIC) framework, designed to facilitate two types of algorithms. The first algorithm allows for the estimation of the benefits of sharing data in a privacy-preserving manner, ensuring that sensitive data is not disclosed. The second algorithm enables the sharing of agreed-upon datasets with specific partners, such as sharing only common attack information. They investigated the practical feasibility of Private Set Intersection (PSI) for predictive IP address blacklisting.

2.3. Bridging the gap

Prior studies have examined the advantages of sharing threat intelligence and the implementation of platforms to facilitate this collaborative process. In addition to these works, other related research has focused on deploying specific tactics and procedures to safeguard sensitive information and prevent its inadvertent disclosure to unauthorized parties. Our primary objective is to bridge a crucial gap in the existing approaches for sharing threat intelligence. Some of the currently proposed techniques are designed to cater to specific types of threat intelligence, while others lack an upfront assessment of the potential usefulness of shared threat intelligence for subsequent correlation analysis. In response to this gap, we endeavor to develop a more comprehensive solution that is not limited by specific threat intelligence types and thoroughly evaluates the value of sharing threat intelligence to facilitate effective correlation analysis.

Our practical contribution aligns with previous findings emphasizing the benefits of sharing actionable threat intelligence while ensuring the secure handling of confidential information. To achieve this, we employ state-of-the-art privacy enhancing techniques that enhance trust in the threat intelligence platform. In line with our goals, our solution offers advanced capabilities for correlation analysis, with a specific emphasis on secure sharing, analysis, and correlation of sensitive information, all while maintaining confidentiality. While we leverage similar techniques as in previous works, we believe we are the first to utilize Private Graph Intersection (PGI) in the context of privacy-preserving sharing and correlation of threat intelligence across multiple organizations. By adopting a graph-based approach, we can effectively represent and share threat intelligence events and also capture the complex relationships and correlations between them.

3. Polyglot persistence and sharing

This study expands on our previous research conducted in the TATIS [7, 8] framework (see Figure 1), utilizing the MISP[5], The Hive, and Cortex[6] threat intelligence platforms, as well as an earlier version of this research [13]. In the forthcoming sections, we will elucidate how we modified this framework to cater to the distinct confidentiality requirements of threat intelligence producers and consumers, enabling them to share and correlate threat events effectively.

[5]https://www.misp-project.org
[6]https://thehive-project.org

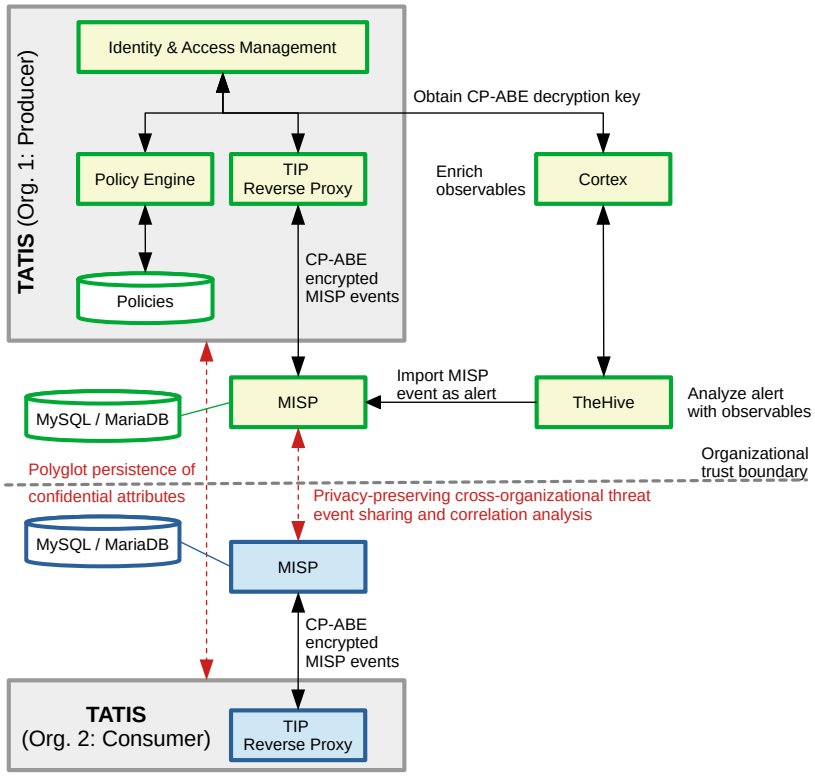


Figure 1: Privacy-preserving sharing and correlation analysis of threat events across the organizational trust boundary.

3.1. Stakeholder concerns and goals for sharing confidential threat intelligence

Threat intelligence producers and consumers often have distinct needs due to their differing roles, responsibilities, and objectives within the cybersecurity ecosystem. Threat intelligence producers are typically organizations or teams responsible for collecting, analyzing, and generating threat intelligence. Their primary focus is on gathering raw data, conducting research, and identifying potential threats. On the other hand, threat intelligence consumers, such as security operations teams or decision-makers, require actionable insights and contextual information to protect their systems and make informed decisions.

Let's examine the example of a simplified MISP threat event, illustrated in Listing 1, where an IP address is depicted as one of the confidential attributes requiring protection. The IP address may uniquely identify a device or network on the internet. If an IP address is associated with a specific victim, it can potentially reveal sensitive information about the target organization or individual, such as their geographical location, network infrastructure, or even specific systems and services in use. Knowledge of the victim's IP address can be valuable to threat actors who may focus their efforts on compromising the victim's systems or exploiting vulnerabilities specific to their network configuration. Last but not least, revealing the IP address of a target victim could tip off adversaries or hinder ongoing security operations. By concealing the IP address, security teams can maintain a strategic advantage in investigating and mitigating threats. Hence, in the context of sharing threat intelligence information, threat intelligence producers and consumers encounter distinct requirements.

3.1.1. Threat intelligence producers

These organizations and teams have primary responsibilities such as collecting and analyzing raw data, identifying potential threats, and generating comprehensive threat intelligence. They actively engage in sharing this intelligence across diverse platforms and communities to enhance collective defense and foster collaboration. However, these organizations may also face specific requirements regarding confidentiality and compliance. As a result, they need

```
{
  "Event": {
    "uuid": "3fdf40c2-7485-11ec-90d6-0242ac120003",
    "date": "2022-01-05",
    "threat_level_id": "1",
    "info": "This is a network threat event",
    "published": true,
    "distribution": "0",
    "Attribute": [{
        "type": "ip-dst",
        "category": "Network activity",
        "to_ids": true,
        "distribution": "5",
        "comment": "This is a sensitive attribute",
        "value": "1.2.3.4",
        "uuid": "da1141b0-712b-4bc8-bf4a-51830f2918c6"
      }, {
        "type": "port",
        "category": "Network activity",
        "to_ids": true,
        "distribution": "5",
        "value": "443",
        "uuid": "61d2ee12-fc7b-4129-8c69-ea856254d923"
      }
    ]
  }
}
```

Listing 1: MISP threat event with 2 attributes in JSON format

mechanisms to ensure controlled access, protect sensitive data, and adhere to legal and regulatory obligations. Consequently, these stakeholders have several options:

1. **Withhold Information**: They can choose not to share certain threat intelligence information at all, preserving its confidentiality and restricting access exclusively to internal teams.
2. **Restricted Sharing**: Alternatively, they can share fully detailed threat intelligence, but limit its dissemination to a restricted set of consumers. This approach allows for more targeted sharing while maintaining confidentiality.
3. **De-identification**: Another option is to share information in a manner that reduces its sensitivity or revealing nature, while still providing value for security analysis. By de-identifying or anonymizing certain aspects, the shared intelligence can protect sensitive details while offering insights and trends.
4. **Combination Approach**: Organizations can also adopt a combination of the above strategies by tailoring their sharing methods based on the specific audience or recipient. They may share fully detailed intelligence with a select group while employing de-identification techniques for broader dissemination.

In summary, these stakeholders have a range of options available to address their confidentiality requirements when sharing threat intelligence. They can choose to withhold information, restrict sharing, de-identify data, or utilize a combination of these approaches to cater to different audiences and maximize the utility of shared intelligence.

3.1.2. Threat intelligence consumers

These stakeholders primarily consist of security operations teams and decision-makers who rely on actionable insights and contextual information to safeguard their systems and make well-informed choices. Their specific needs revolve around obtaining a concise and filtered view of threat intelligence, focusing on the risks and vulnerabilities that directly pertain to their systems or operations. Upon receiving threat intelligence, these stakeholders process it for various purposes:

1. **Incident Insights and Response**: They analyze the received threat intelligence to gain deeper insights into a specific incident. This helps them understand the nature of the threat and formulate effective response strategies to mitigate its impact.
2. **Confirmation of Incidents**: They verify the occurrence or sightings of threats mentioned in the intelligence. Confirming the validity of these threats allows them to prioritize and allocate resources accordingly.

3. **Enhanced Threat Analysis**: The received threat intelligence can augment the analysis of existing threat events. By incorporating additional information, they can refine their understanding of the threats and their potential implications.
4. **Correlation with Local Observations**: They seek to correlate the incident mentioned in the intelligence with other threat events observed locally. This correlation enables them to identify patterns, uncover potential connections, and gain a comprehensive understanding of the overall threat landscape.

In summary, these stakeholders utilize the received threat intelligence to gain valuable insights into specific incidents, confirm threats, enhance threat analysis, and establish correlations with locally observed events. This information empowers them to effectively respond to incidents, allocate resources appropriately, and maintain a robust security posture.

3.1.3. Limitations of state-of-practice sharing protocols

The Traffic Light Protocol (TLP)[7] is a tagging scheme and framework used for sharing sensitive information, primarily within the cybersecurity community. While TLP is widely utilized and generally effective, it does possess certain limitations.

The TLP employs color-coded designations (e.g., Red, Amber, Green) to signify the sensitivity and sharing restrictions of information. However, different organizations or individuals may interpret these colors differently, resulting in potential confusion or miscommunication. The absence of clear guidelines can lead to disparities in the understanding and implementation of TLP markings. Additionally, the TLP offers a broad categorization of information sensitivity but lacks finer levels of granularity. This limitation may make it inadequate for situations where information falls between the defined TLP levels or necessitates more nuanced sharing restrictions. The absence of intermediate designations may hinder precise communication and appropriate handling of sensitive data. Furthermore, the TLP relies on trust between parties involved in information sharing. While the protocol encourages responsible handling of sensitive information, there is no guarantee that the receiving party will strictly adhere to the prescribed sharing restrictions or adequately safeguard the information as intended. This reliance on trust introduces a level of uncertainty and potential risk when sharing sensitive data.

It is essential for organizations to recognize these limitations of the TLP. These limitations highlight the necessity for a more adaptable threat intelligence persistence and processing layer. Additionally, cryptographic methods are required to effectively enforce access constraints.

3.2. Polyglot persistence of confidential attributes

In contrast to the example provided in Listing 1, a typical MISP threat event encompasses numerous attributes and may even incorporate multiple MISP objects with multiple annotations. Many of these attributes contain confidential or private information. Our framework offers the capability to selectively filter, transform, and encrypt these attributes as needed. The underlying relational database within MISP (e.g. MySQL or MariaDB) is designed with strong typing, allowing for on-the-fly modification of attribute types. This means that instead of storing the original IPv4 address, our framework can store a larger base64 encoded payload, which can be either plaintext or ciphertext. This enables the secure storage of sensitive information within the MISP persistence layer. One advantage of leveraging MISP's persistence layer is that no additional functionality is required to share threat intelligence, whether it is plaintext, encrypted or privatized, with other MISP instances across the organizational trust boundary. This sharing can be achieved through MISP's push or pull synchronization mechanisms, allowing for seamless and secure dissemination of threat intelligence between interconnected MISP instances. This integration simplifies the process of sharing encrypted or privatized threat intelligence without the need for extra infrastructure or complex communication channels.

Initially, TATIS [7, 8] provided fine-grained access control to threat intelligence using Ciphertext-Policy Attribute-Based Encryption (CP-ABE). In the case of the event example presented in Listing 1, the original IP address attribute was encrypted using AES, and the AES secret key was protected with CP-ABE. The decision to utilize CP-ABE to protect the AES secret key was motivated by the fact that AES secret keys are typically shorter than most attribute

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

values. This characteristic allows for faster CP-ABE decryption of the AES secret key. Once decrypted, the AES secret key can be used to decrypt the actual threat intelligence using AES. However, it is important to note that AES encryption as a privacy-enhancing technique presents a drawback. It hinders the analysis of correlations between events and attributes unless the consumer possesses the capability to decrypt the protected information. Specifically, the consumer must possess a CP-ABE decryption key constructed with user profile attributes that satisfy the conditions of the encryption policy used to encrypt the AES secret key.

To enhance our solution's capabilities, we have incorporated polyglot functionality, allowing for the storage of multiple variations of the same attribute. This enables the restriction of access to sensitive attributes at different levels of granularity, while still supporting correlation analysis for threat intelligence consumers who do not possess decryption keys. The different variations include:

- **Plaintext**: The threat event and its associated attributes are stored in clear, without any encryption or obfuscation. This variation is suitable for insensitive or public information that does not require access restrictions. An example of this is TLP:WHITE threat intelligence.
- **Suppression**: Attributes that are deemed too business sensitive, private, or involve personally identifiable information are entirely removed from the stored data. This ensures that such information remains confidential and is not accessible to threat intelligence consumers.
- **Transformed**: Instead of exposing the original attribute, it undergoes one or more transformations to fulfill the needs of threat intelligence consumers without revealing sensitive details. These transformations can include techniques like hierarchical encoding, hashing, or other suitable methods that reduce the level of detail while maintaining the attribute's interpretability and correlation potential.

In terms of transforming sensitive threat attributes, there are various transformation techniques that can be employed to restrict the amount of information revealed and control access to the information. These techniques include:

- **Hierarchical Encoding**: The value of an attribute is transformed and generalized using a predefined hierarchy. This reduces the uniqueness of the value, making it less exact for correlation with other events and attributes. However, the transformed attribute retains semantic interpretability.
- **Hashing**: The original attribute is replaced with its hashed counterpart. Different hashing schemes can be utilized, such as secure hashes (e.g., SHA-256), password hashes with salt and iteration count (e.g., PBKDF2), hash-based message authentication codes (e.g., HMAC-SHA-256), fuzzy hashing (e.g., ssdeep), Bloom filters, and more. Attribute correlation remains possible, but the content interpretation is sacrificed.
- **Encryption**: Individual attributes can be encrypted using CP-ABE (Ciphertext-Policy Attribute-Based Encryption). Each attribute can be encrypted with a different encryption policy, ensuring that only individuals possessing a matching decryption key can retrieve the AES secret key and use it to obtain the original plaintext. This technique limits attribute correlation and interpretation to a restricted and authorized audience.
- **Hybrid**: Hidden values of a set of attributes are employed in a key derivation scheme (e.g., PBKDF2) to generate a secret key. This key is then utilized to protect an attribute. As a result, only threat consumers who possess knowledge of the hidden values can compute the derived key to decrypt the encrypted attribute.

Bloom filters, invented by Burton Howard Bloom in 1970 [28], are probabilistic space-efficient data structures for efficiently testing the membership of an element in a set. A Bloom filter works by using a fixed size bit array of zeros and a set of hash functions. When inserting an element into the filter, the element is hashed multiple times, and the corresponding positions in the bit array are set to 1. To check if an element is in the set, it is also hashed using the same hash functions, and the filter checks the corresponding positions in the bit array. If all of these positions are 1, the element is likely in the set, but false positives are possible. However, if any of the positions are 0, it means that the element is definitely not in the set. In our research, the Bloom filter allows a producer to share information regarding its threat events, and consumer to verify whether they have any events in common. Importantly, this verification process is designed to prevent the consumer from executing brute force queries on the filter to gather new information, as doing so will lead to false positives.

These transformation techniques can be applied at the attribute level and can also be combined. For instance, the same attribute information can be provided in both encrypted form (supporting in-depth analysis for those with a decryption key) and hashed form (enabling correlation analysis with other events for threat event consumers without a decryption key). The appropriateness of these techniques in terms of security and privacy varies based on the attribute type and their combination for polyglot persistence. When an attribute is limited to a small set of values, it becomes practically possible to pre-compute all (unsalted) SHA-256 hashes, which makes the CP-ABE encryption of the same attribute pointless. Likewise, storing an attribute in both plaintext and encrypted form is also futile. By employing these polyglot capabilities, our solution enables the storage of different variations of attributes, allowing for controlled access and analysis of threat intelligence. It caters to the requirements of diverse threat intelligence consumers and producers, providing the appropriate level of granularity and protection for sensitive information, allowing for different levels of analysis and correlation based on the authorized audience's capabilities and permissions.

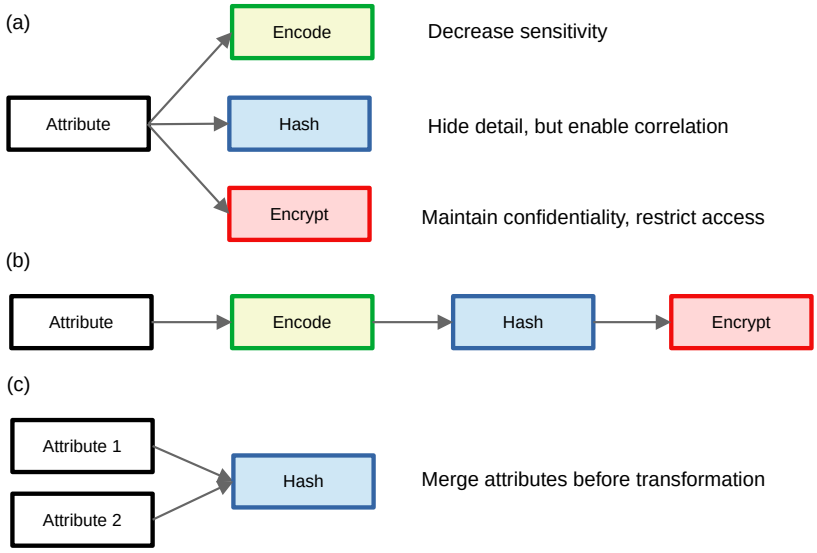


Figure 2: A variety of attribute transformation pipelines

3.3. Composition of transformation techniques

The transformation techniques discussed earlier can be combined or arranged in various ways, offering flexibility in protecting sensitive threat attributes. Figure 2 illustrates different scenarios showcasing the possible combinations:

- In scenario (a), the original attribute value is not shared in its clear form. Instead, it is protected using three distinct methods, each serving different purposes for diverse audiences of threat intelligence consumers, as described in the previous section.
- For scenario (b), a two-step transformation is depicted. In the first step, individual continuous valued attributes are generalized into an interval-based hierarchy and then top- or bottom-coded. In the second step, the top- or bottom-encoding is transformed using HMAC-SHA-256. This approach hides the structure of the encoding hierarchy itself while enabling correlation of attributes within the same interval. Lastly, the hashed attributes are encrypted with CP-ABE, restricting access solely to consumers authorized according to the encryption policy.
- Scenario (c) is particularly useful when it is feasible to pre-compute the (unsalted) hashes for all possible values of a single attribute type. This allows for the original attribute value to be learned by matching the hashes. However, for joint attribute values (e.g., combining IP address and network port), creating a lookup table of hashes for all possible combinations may not be practically feasible. Nonetheless, the hash of the joint attributes can still be utilized to analyze specific correlations.

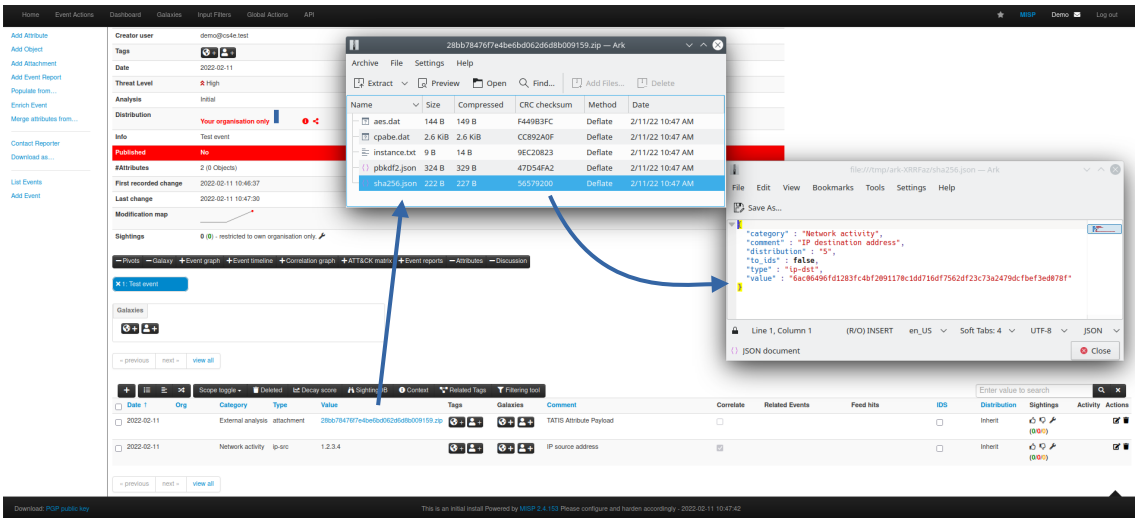


Figure 3: A MISP threat event with (a) a plaintext attribute `ip-src` and (b) an attribute `ip-dst` stored as a ZIP file containing the CP-ABE, SHA-256 and PBKDF2 transformations

By considering these scenarios and utilizing appropriate combinations of transformation techniques, organizations can effectively protect sensitive threat attributes while retaining the ability to perform correlation analysis and extract valuable insights from the threat intelligence data.

In addition to attribute-level protection, our framework also extends its support to anonymizing threat intelligence at the event level. This approach not only mitigates the disclosure of sensitive attributes but also addresses the risk of membership inference attacks. We have incorporated well-known privacy-enhancing techniques to achieve this, including k-anonymity, l-diversity, t-closeness [29], and differential privacy [30]. To implement these techniques effectively, our framework requires a list of attribute types (known as quasi-identifiers) that should be considered for potential identification and anonymization. The methods, such as k-anonymity, ensure that the released data cannot be distinguished or linked to specific individuals or entities. For instance, in the case of achieving k-anonymity, the IP address ”1.2.3.4” from Listing 1 may be hierarchically encoded to ”1.2..”, preserving privacy while still allowing analysis at a broader level.

It is important to note that the anonymization techniques mentioned earlier are commonly employed when releasing datasets to the broader community. However, in the context of threat intelligence sharing, the approach is different. Here, threat events are processed and shared individually and incrementally, rather than being released as complete datasets. This distinction is crucial because the nature of threat intelligence requires real-time analysis and sharing of specific threat events as they occur.

3.4. Policy-driven polyglot persistence

The configuration of polyglot persistence for threat intelligence is driven by policies. An example of such a policy is presented in Listing 3 in the Appendix. This policy, set by the security administrator, determines the sharing of sensitive, private, or confidential threat intelligence with third parties. It allows for the flexible combination of privacy-enhancing techniques, as depicted in Figure 2. Additionally, the polyglot solution treats the creation of a new MISP threat event by a security analyst or the enrichment of an existing event with an attribute in the same manner. Thus, it supports the entire lifecycle of threat intelligence.

With the specific policy in place, we process attributes of types `ip-dst` and `email`, which belong to the default attributes and categories provided by MISP out-of-the-box[8]. The same process is applied to attributes of a MISP object created based on a MISP object template[9]. In this case, the policy demonstrates (a) the transformation of individual attributes using one or more privacy-enhancing techniques (PETs), and (b) the application of k-anonymity

[8]https://www.misp-project.org/datamodels/

[9]https://www.misp-standard.org/rfc/misp-standard-object-template-format.html

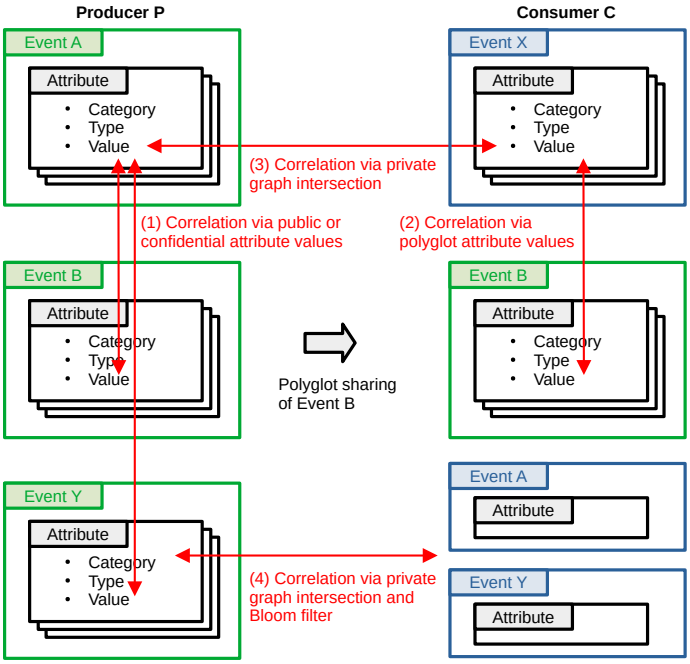


Figure 4: Intra- and cross-organizational correlation of threat events with confidential or sensitive attribute values

to each MISP object instantiated from the `custom_network_security_object` template before attribute transformations occur. This anonymization step is performed prior to storing the threat event in MISP's relational database (i.e. MySQL or MariaDB) and potentially sharing it with MISP instances from other organizations.

Figure 3 illustrates the practical implementation of the framework within MISP's dashboard. The figure showcases a simulated threat event with two attributes: the `ip-src` attribute, which is stored and shared in plaintext, and the `ip-dst` attribute, which is protected according to the policy outlined in Listing 3. For the latter attribute, the original value is disclosed in three different ways, including CP-ABE encryption and hashing with SHA-256 and PBKDF2. The three versions of the attribute are bundled in a ZIP file, and Figure 3 also displays the list of files within the ZIP archive and the contents of the SHA-256 variant of the `ip-dst` attribute. Instead of consolidating all polyglot variants of an attribute in a ZIP file, our solution can also store each variant individually in MISP, using the '*Comment*' field to preserve metadata of the polyglot variant. This approach allows for customized distribution levels per variant.

4. Privacy-preserving correlation via private graph intersections

The Threat Intelligence Platform (TIP) plays a crucial role in assisting security analysts by providing actionable information on adversaries. It accomplishes this by characterizing attack campaigns and establishing detection patterns to effectively defend against cyber attacks, potentially in an automated manner and with minimal delay. TIPs typically achieve this by comparing internal and third-party threat events from intelligence feeds, identifying correlations between these events, and analyzing the attributes they share. For instance, if two threat events targeting different entities share a common attribute indicating the source IP address of the attacks, it suggests that the same adversary may be responsible for both campaigns.

Our framework facilitates the creation of a correlation graph that spans multiple organizations while preserving privacy and confidentiality. It focuses specifically on sensitive or confidential attributes owned by different organizations, which are not shared with other threat consumers in any form, be it original or derived (e.g., encoded, hashed, encrypted). In this context, organizations are willing to collaborate in establishing connections between their respective threat events and incidents without compromising the confidentiality of the sensitive threat intelligence that underlies these connections.

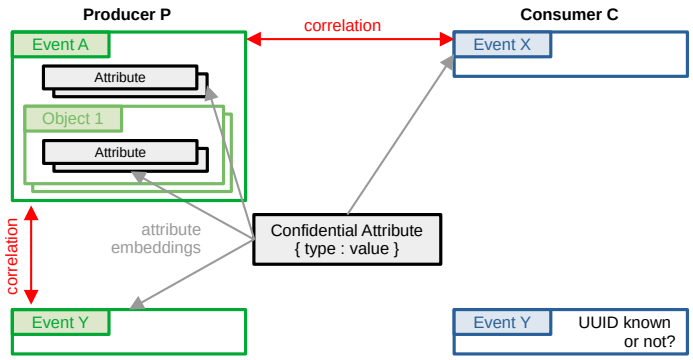


Figure 5: Embedding of confidential attributes in threat events and their objects, inducing correlations between them

4.1. Correlation graphs of threat events

The correlation graphs in MISP provide valuable insights into the relationships between threat events and the attributes they share. These graphs help security analysts identify clusters of activities, attack campaigns, and enable them to navigate from one threat event to another. Correlations occur at the attribute level, where attributes are embedded within different threat events. This is illustrated in Figure 4. These correlations can be established based on the original attribute contents or the polyglot variants of the attributes, as explained in Section 3.2 and depicted in Figure 2:

- Exact or fuzzy match of original attribute values
- Exact match of hierarchically encoded attribute values
- Subsumption of hierarchically encoded attribute values
- Exact match of hashed attribute values
- Exact match of hashed attribute value combinations

The illustration represents two entities: a threat intelligence producer, denoted as P, and a threat intelligence consumer, denoted as C. The producer has knowledge of three threat events, namely A, B, and Y, while the consumer is aware of events A, X, and Y. A polyglot variant of Event B is shared specifically with consumer C. Both parties are aware of events A and Y, but the producer, P, possesses additional confidential attributes related to these events that are not shared with or known to consumer C. In this context, non-shared attributes refer to those whose values have not been exchanged in any manner between the two parties. However, it is possible for producer P and consumer C to independently gather confidential attributes with matching types and values. These confidential attributes may be embedded in different threat events, as depicted in Figure 5.

Figure 4 presents four scenarios, indicated by the red text and labeled as (n), where n $\in$ [1..4]. Scenario (1) represents the default mode of MISP, where attribute values within threat events A, B, and Y are locally correlated in detail. This includes attributes that are pulled or pushed from other MISP instances. Scenario (2) utilizes the correlation capabilities of MISP between events X and B. However, in this case, the correlation is performed against the polyglot variant of the shared (and potentially privatized) attributes of event B. Scenario (3) aims to correlate non-shared attributes between threat events A and X across the producer P and consumer C. To achieve this, a private graph intersection (PGI) protocol and Bloom filters are employed. These techniques enable learning about correlated events without disclosing sensitive or confidential attribute values that are unknown to the other party. Scenario (4) focuses on a threat consumer C discovering a correlation between two threat events, A and Y. Both events are known to both parties, but only the threat producer P is capable of establishing the correlation due to non-shared attributes that remain undisclosed to the threat consumer C.

Scenarios (1) and (2) can leverage existing functionalities within MISP for matching transformed attributes and constructing the correlation graph. On the other hand, scenarios (3) and (4) require the utilization of a private graph

Algorithm 1 Hash attributes based on their type and value

```
procedure HashAttr(attributes)
    l ← List()
    for a ∈ attributes do
        h ← hash(a.type : a.value)                ▷ 128-bit hash
        l.add(h)
    end for
    return l
end procedure
```

Algorithm 2 PSI of hashed confidential attributes

```
procedure PSIConfAttr(p, c)                       ▷ producer and consumer
    s1 ← HashAttr(p.get_confidential_attributes())
    s2 ← HashAttr(c.get_confidential_attributes())
    s ← s1 ∩ s2                                   ▷ private set intersection [12]
    return s
end procedure
```

intersection (PGI) protocol and Bloom filters to enable correlation discovery without revealing sensitive or confidential attribute values that are unknown to the respective party. Further details on these last two scenarios will be provided in the next subsection.

4.2. Correlating non-shared confidential attributes with private graph intersection

To analyze scenarios (3) and (4), we will examine the correlation graphs of the threat intelligence producer, denoted as P, and the threat intelligence consumer, denoted as C. Both parties possess private graphs that consist of events containing non-shared confidential attributes, which they do not disclose to other entities. We will represent these graphs as $G_P = (V_P, E_P)$ and $G_C = (V_C, E_C)$, where V and E represent the lists of vertices and edges in the graphs, respectively. The vertices in the graphs represent threat events that include confidential attributes, and the edges represent correlations between these events.

In scenario 3, the threat intelligence consumer C discovers unknown correlations between threat events in two situations. First, if consumer C and producer P share a confidential attribute within their respective threat events, denoted as $v_{C,i} \sim v_{P,i'}$, a correlation is established. Second, if producer P has two threat events, $v_{P,i} \sim v_{P,j}$, within V_P, and these events share a confidential attribute. Consumer C is also aware of these two threat events, $v_{C,i'}$ and $v_{C,j'}$, but is unaware of the correlation induced by the shared confidential attribute.

4.2.1. Scenario 3: Cross-organizational correlation

(Step 1) The threat intelligence producer P and the threat intelligence consumer C perform the computation of the private set intersection (PSI_1)[31] on their respective sets of non-shared confidential attributes. Specifically, they compute the 128-bit hash of the type and value fields of these attributes, as shown in Algorithms1 and 2. By using the hash, we address the variation in attribute value sizes, which can range from a few bytes (e.g., an IP address) to several megabytes (e.g., a malware sample). Algorithm 2 provides a simplified representation of the PSI algorithm. Within our framework, we leverage the Low Multiplicative Complexity (LowMC) PSI implementation developed by Kales et al. [12] for its performance advantages. The producer P and the consumer C can reconstruct the original contents of the confidential attributes from the hashes obtained during the PSI computation.

(Step 2) Subsequently, the threat intelligence producer P and the threat intelligence consumer C individually construct their correlation graphs, denoted as G_P and G_C respectively, based on the threat events that are influenced by the confidential attributes identified in the PSI_1 computed in step 1. They then proceed to compute the private graph intersection (PGI) of the event correlation graphs G_P and G_C. The intersection of both graphs is defined as $G_I = (V_I, E_I) = G_P \cap G_C$, where $V_I = V_P \cap V_C$ represents the common vertices, and $E_I = E_P \cap E_C$ represents the

Algorithm 3 Bloom filter of attribute embeddings for correlations not in PGI (step 2) for attributes in PSI_1 (step 1)

```
1: procedure AttrEmbeddings(attributes)                    ▷ attributes in PSI_1
2:     b ← BloomFilter()
3:     for a ∈ attributes do
4:         events ← a.get_events()
5:         for e ∈ events do
6:             h ← hash(a.type : a.value : e.uuid)          ▷ 128-bit hash
7:             b.add(h)
8:         end for
9:     end for
10:    return b
11: end procedure
```

common edges. The correlations can be expressed using an edge matrix E_P or E_C, as depicted below:

$$E = \begin{pmatrix} e_{1,1} & \cdots & e_{1,m} \\ e_{2,1} & \cdots & e_{2,m} \\ \vdots & \ddots & \vdots \\ e_{m,1} & \cdots & e_{m,m} \end{pmatrix} \quad \text{with } e_{i,j} = 1 \text{ iff } V_i, V_j \text{ correlated}$$

The value of edge $e_{i,j}$ is 1 if there exists a common attribute between threat events V_i and V_j that was identified during the PSI1 computation in step 1. Otherwise, the value is 0. Upon calculating the PGI, both the threat intelligence producer P and the threat intelligence consumer C are aware of the shared threat event correlations. It is important to note that the size of the edge matrix Em, m may differ for each party, and they may have different sets of threat events. Furthermore, the edge matrix E is typically sparse, with only a few 1's and many 0's. Consequently, both parties encode each correlation (i.e., an edge with $e_{i,j} = 1$) between threat events V_i and V_j by computing the 128-bit hash of their UUIDs ($hash(V_i.uuid : V_j.uuid)$). To ensure symmetry, V_i and V_j are sorted based on their UUIDs before the hash computation. Producer P and consumer C then perform a similar PSI_2 computation on both sets to identify the common correlations or edges E_I of the PGI.

In *(Step 3)*, the threat intelligence producer P identifies the confidential attributes responsible for threat event correlations that were not found in the PGI (i.e., PSI_2) computed in step 2. These are the correlations that producer P is willing to share. To accomplish this, P creates a Bloom filter using Algorithm 3, which stores the embeddings of these attributes within their respective threat events.

Once the Bloom filter is constructed, the threat intelligence consumer C can proceed to iterate through all the confidential attributes in the PSI_1 obtained from step 1. For each attribute, C iterates through the UUIDs of its own threat events to check if the hash value *hash(a.type:a.value:e.uuid)* is present in the Bloom filter. If a match is found, C has learned a new correlation with a threat event that does not yet have this confidential attribute.

It is important to note that a Bloom filter may result in false positives, but not in false negatives. The false positive rate ϵ depends on factors such as the number of bits m in the Bloom filter, the number of elements n to be stored, and the number of hash functions k used.

$$\epsilon \approx (1 - e^{-\frac{nk}{m}})^k$$

with, for a given m and n, the value k that minimizes ϵ:

$$k = \frac{m}{n} ln2 = -log_2 \epsilon$$

Producer P can decide which ϵ value it finds appropriate.

4.2.2. Scenario 4: Remote-organizational correlation

In the given scenario, the threat intelligence producer P has identified a correlation between two confidential attributes with the same type and value. These attributes are embedded in two different threat events, $v_{P,i}$ and $v_{P,j}$,

Algorithm 4 Bloom filter of correlated events for non-shared confidential attributes not in PSI_1 (step 1) for events in PSI_3 (step 5)

```
1:  procedure EventCorr(attributes)                          ▷ attributes not in PSI_1
2:      b ← BloomFilter()
3:      for a ∈ attributes do
4:          events ← a.get_events() ∩ PSI_3                  ▷ Only events in PSI_3
5:          for e_1, e_2 ∈ events do
6:              h ← hash(e_1.uuid : e_2.uuid)                ▷ 128-bit hash
7:              b.add(h)
8:          end for
9:      end for
10:     return b
11: end procedure
```

within the set of threat events V_P. On the other hand, the threat intelligence consumer C is aware of two threat events, $v_{C,i'}$ and $v_{C,j'}$, within its set of threat events V_C, but it does not possess the confidential attribute embedded in these events. Therefore, consumer C has no knowledge of the correlation between these two events.

Producer P is willing to share the correlation with consumer C at the level of the threat events, without revealing the contents of the confidential attributes. However, this sharing will only occur if consumer C already has knowledge of the threat events in the first place.

In *(Step 4)*, when the confidential attribute is part of the PSI_1 computed in step 1 of scenario 3, consumer C will have another threat event, vC, k', within its set of threat events V_C. This additional event, as part of scenario 3, enables consumer C to learn about the correlation between $v_{C,k'} \sim v_{C,i'}$ and $v_{C,k'} \sim v_{C,j'}$. With this information, consumer C can independently infer and learn about the correlation between $v_{C,i'} \sim v_{C,j'}$.

In *(Step 5)*, when the confidential attribute is not part of the PSI_1 computed in step 1 of scenario 3, meaning it is only known to producer P and not to consumer C, consumer C will not learn about the correlation through the protocols of scenario 3. In this case, both parties proceed to compute the PSI_3 of the UUIDs of all their non-shared events. If sharing is symmetric, the UUIDs of shared events would already be known. Producer P then computes a Bloom filter specifically for correlated threat events where the correlation is induced by non-shared confidential attributes that are not present in the PSI_1 computed in step 1 (see Algorithm 4). This is done for those common threat events found in the PSI_3.

In *(Step 6)*, consumer C learns about unknown correlations by performing pairwise hashing between the event UUIDs it has in common with producer P within the PSI_3. Consumer C checks whether these pairwise hashes, computed as $hash(e_1.uuid : e_2.uuid)$, are present in the Bloom filter constructed by producer P. By doing so, consumer C can identify and learn about unknown correlations between threat events, even when the confidential attributes are not directly shared.

5. Evaluation

In this section, our focus will be on thoroughly assessing the effects on security and performance when connecting non-shared confidential attributes with the private graph intersections and Bloom filters discussed earlier in the previous section.

5.1. Security evaluation

For each of the 6 steps in scenarios 3 and 4, we will review the security impact:

- *Step 1:* By performing PSI calculations on the hashes of confidential attributes, the embedding of these attributes within their corresponding threat events remains concealed. Put simply, one party does not gain knowledge of the association between confidential attributes and the threat events they are embedded in at the other party.

- *Step 2:* Once the protocol concludes, both parties can determine the shared threat event correlations and identify the correlations that are exclusive to each party. However, the confidential attributes that led to these correlations at the other party remain undisclosed. Neither party has knowledge of the other's threat events, except for those included in the PGI. It is worth noting that an attribute with the same type and value, but lacking the confidential designation, is not included in the PGI. Therefore, it does not reveal any additional information to the other party.

- *Step 3:* The Bloom filter guarantees that consumer C remains unaware of the embeddings of confidential attributes for threat events it is not aware of. To access such information, C would need to correctly guess the UUID, which is a random 128-bit value. Even if C were lucky enough to make a correct guess, this identifier does not reveal any additional information about the unknown threat event. Additionally, attempting a brute force attack would result in mismatches, as a Bloom filter is susceptible to false positives.

- *Step 4:* Consumer C gains knowledge about additional correlations by utilizing the reflexive and transitive properties of correlations. However, this step does not introduce any additional security implications.

- *Step 5:* Both parties acquire knowledge of the UUIDs corresponding to the threat events they share, without gaining any additional information beyond that.

- *Step 6:* Consumer C becomes aware of new correlations among the threat events it was already aware of. However, C does not gain any knowledge regarding the confidential attribute of producer P that caused these correlations, nor does it acquire information about correlations involving other threat events unknown to C.

The analysis of correlations between threat events relies heavily on the security properties of PSI [12] and the probabilistic characteristics of Bloom filters. By performing PSI_1 on the confidential attributes, the connection to the corresponding threat events remains undisclosed. Similarly, PSI_2 for computing the private graph intersection avoids revealing the relationship with the confidential attributes. Moreover, PSI_3 on the UUIDs of the threat events does not leak any additional information about the events themselves.

Consumer C discovers the correlations through querying a Bloom filter. However, brute forcing such queries is computationally intensive and may lead to false positives, which is controlled by the design parameter ϵ determined by the threat intelligence producer P.

A malicious consumer C may attempt to extract information from producer P by initiating a forged PSI using carefully selected fake confidential attributes. The adversary computes hashes of counterfeit confidential attributes, which may reveal the existence of these attributes at the producer but not their embeddings within threat events. The adversary might also attempt to learn about correlations between threat events, but this would require hashing every possible combination of UUIDs. Likewise, the probabilistic nature of a Bloom filter prevents the adversary from brute forcing all attribute and threat event combinations.

Feed	URL	Events	Attributes	Unique Attributes
1. CIRCLE OSINT Feed	https://www.circl.lu/doc/misp/feed-osint/	1485	487751	345197
2. The Botvrij.eu Data	https://www.botvrij.eu/data/feed-osint/	332	19824	19439
3. ThreatFox	https://threatfox.abuse.ch/downloads/misp/	794	1017477	1012989
4. MalwareBazaar	https://bazaar.abuse.ch/downloads/misp/	725	3399921	2461849
5. URLhaus	https://urlhaus.abuse.ch/downloads/misp/	772	31736027	16634935

Table 1: Community driven threat intelligence feeds, the number of threat events and attributes (on May 20, 2023).

5.2. Performance impact for simulated scenarios

In the performance evaluation, we utilize the threat intelligence feeds listed in Table 1. These feeds were collected on May 20, 2023. For example, the OSINT feed consists of 1485 threat events. Each event typically includes a set of attributes (or type-value pairs), which may also include MISP objects characterized by their own attributes. Taking all attribute types into account, the feed comprises 487751 attributes (i.e. type-value pairs), with 345197 unique type-value pairs. The other threat intelligence feeds have different characteristics.

5.2.1. Experimental setup

The performance benchmark experiments presented below simulate extreme scenarios in terms of the number of confidential attributes, resulting in high computational complexity. These experiments were conducted on a system equipped with an 11th Gen Intel Core i7-11800H CPU operating at 2.30GHz and 32GB of memory. After merging the two feeds, we set up a MISP instance and our framework on separate virtual machines, allocated 10GB of memory and 4 virtual CPU cores for each. Both instances are configured with a subset of threat events selected through random subsampling.

Experiment 1. For this experiment, we will utilize the initial two threat intelligence feeds: the CIRCLE OSINT and Botvrij.eu Data feeds. These feeds collectively contain approximately 1817 events and 361640 unique attributes. Among these attributes, there are 2996 instances where the same type-value pair appears in both feeds, indicating duplicate attributes. Moreover, there are 358644 type-value pairs that occur only once. To proceed with the experiment, we will randomly select 1000 threat events from each feed for both the threat intelligence producer (P) and the consumer (C). Since the total number of events available is 1817, it is expected that there will be some overlap. Indeed, 544 threat events are in common between P and C.

In this experimental setup, both parties will independently choose 1000 type-value pairs from the available set. These selected attributes will be designated as confidential. Among the chosen pairs, 10% will be derived from the 2996 duplicate attributes, while the remaining 90% will be selected from the 358644 attributes pairs that occur only once. Two key observations can be made based on this experimental setting:

- There is no assurance that the randomly selected 1000 attributes (or type-value pairs) will be present in any of the chosen 1000 threat events. The selection of type-value pairs and the occurrence of threat events are independent of each other.
- Considering that the combined threat intelligence feeds contain over 350000 distinct type-value pairs, the likelihood of the producer P and the consumer C sharing common confidential type-value pairs is relatively low.

	Confidential attributes	**Common confidential attributes**	**Confidential attribute embeddings**		**Unique associated events**	
			Producer	Consumer	Producer	Consumer
Experiment 1.a	1000	7	803	678	214	220
Experiment 1.b	2000	17	1701	1399	302	339
Experiment 1.c	5000	126	4308	3284	498	508
Experiment 1.d	10000	574	10700	6492	638	633

Table 2: In experiment 1, producer P and consumer C have each randomly selected *1000 threat events (544 in common)*, and designated a growing number of attributes as confidential.

The characteristics of the threat event subsets of producer P and consumer C are presented in Table 2. From the 1000 randomly selected threat events, the producer P and consumer C have 544 threat events in common (i.e. they share the same UUID), meaning that both of them have 456 unique events.

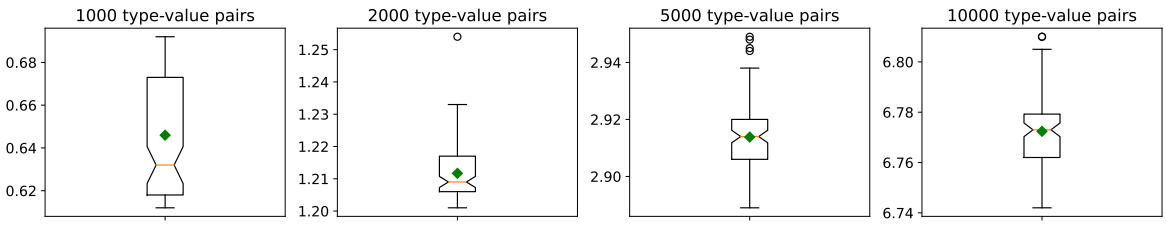
In experiment 1.a, where each party independently selects 1000 confidential type-value pairs at random, both parties have 7 confidential type-value pairs in common. Among these pairs, producer P observes 803 instances of one of the confidential key-value pairs within its 1000 threat events. These 803 occurrences are associated with 214 distinct threat events. Consumer C exhibits similar observations, i.e. 678 occurrences within its 1000 threat events associated with 220 distinct threat events. As the number of confidential type-value pairs increases (i.e. experiments 1.b, 1.c and 1.d), the number of common type-value pairs between the producer P and consumer C also tends to increase. Furthermore, with a greater number of occurrences of these type-value pairs in their respective threat events, the count of unique associated events also rises. Considering that the number of unique threat events is significantly lower (4 to 15 times) than the number of confidential attribute embeddings, it indicates that multiple confidential attributes are present within the same threat events.

Figure 4 showcases four scenarios that are crucial for sharing and analyzing threat intelligence. These scenarios provide valuable insights and facilitate the examination of threat intelligence data. Scenarios (1) and (2) represent

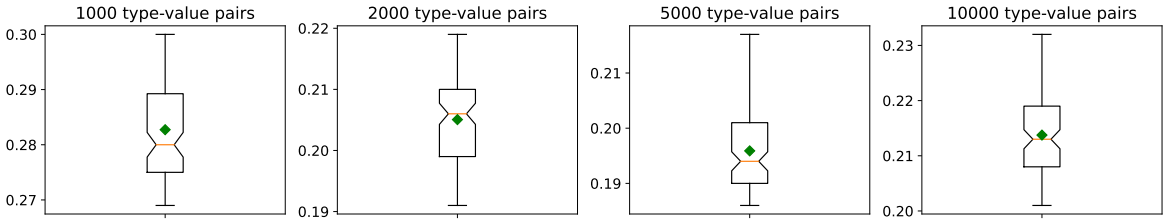
correlation analyses that are already being conducted by MISP. For the rest of our experiments, our attention will be directed towards the cross-organizational correlation of threat events with confidential or sensitive attribute values. To evaluate the privacy-preserving correlation analysis for scenarios (3) and (4), we introduce a random removal of attributes within the 544 threat events that are shared between the producer P and consumer C. Specifically, we delete 50% of the occurrences of producer P's confidential attributes in the common threat events of consumer C. This deletion provides an opportunity for consumer C to learn the correlations that were previously unknown due to the removed attributes.

	Confidential attributes	**Common confidential attributes**	**Confidential attribute embeddings**		**Unique associated events**	
			Producer	Consumer	Producer	Consumer
Experiment 2.a	1000	10	1078	1501	282	313
Experiment 2.b	2000	20	2472	2256	502	481
Experiment 2.c	5000	155	8720	5869	723	763
Experiment 2.d	10000	560	13669	11433	945	879

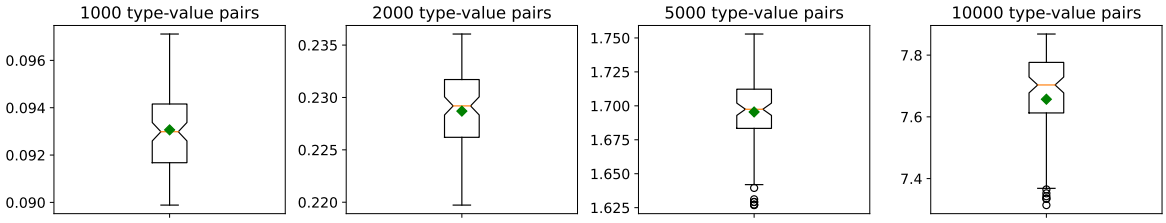
Table 3: In experiment 2, producer P and consumer C have each randomly selected *1500 threat events (1246 in common)*, and designated a growing number of attributes as confidential.



(a) Private Set Intersection 1 (step 1)

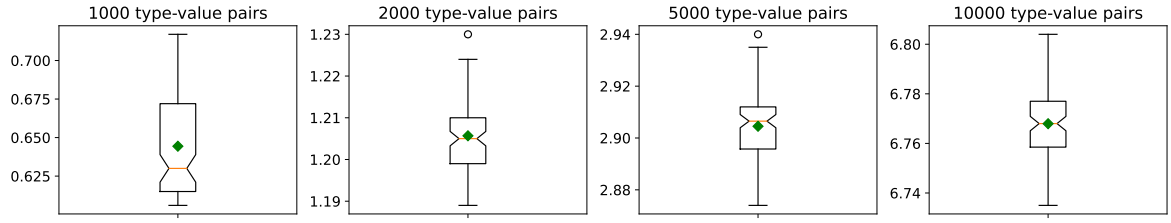


(b) Private Graph Intersection 2 (step 2)

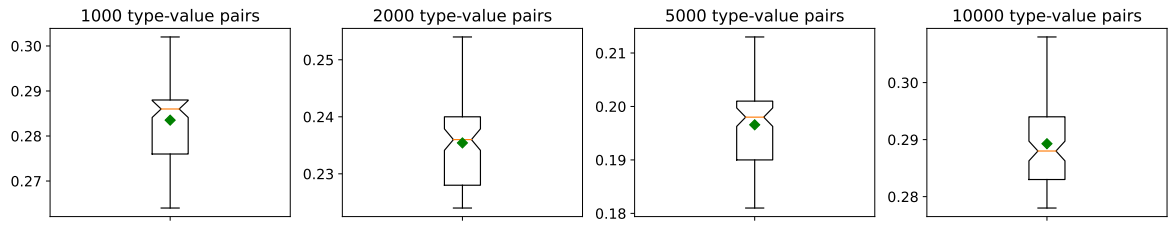


(c) Bloom filter test (step 3)

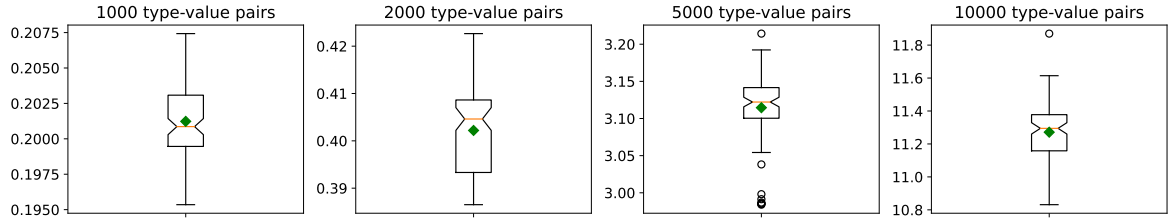
Figure 6: Amount of time spent (in seconds) for each of the different steps in the algorithms of scenario (3) for producer P and consumer C each having 1000 random threat events.



(a) Private Set Intersection 1 (step 1)



(b) Private Graph Intersection 2 (step 2)



(c) Bloom filter test (step 3)

Figure 7: Amount of time spent (in seconds) for each of the different steps in the algorithms of scenario (3) for producer P and consumer C each having 1500 random threat events.

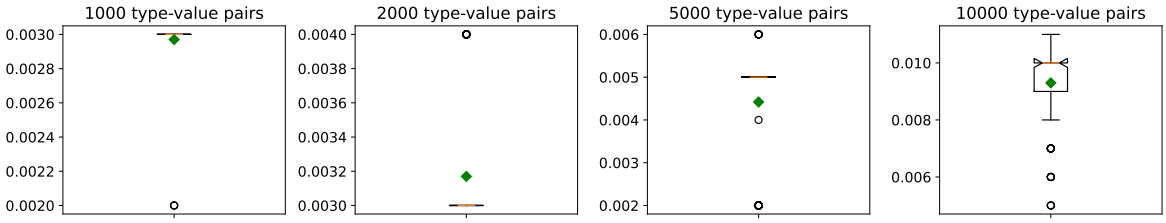
Experiment 2. A second larger experiment will randomly select for producer P and consumer C each having 1500 random threat events. The details are depicted in Table 3. Here, it is important to note that in experiment 2.d, both the producer P and consumer C have a significantly larger number of associated events for their respective confidential attributes. Specifically, there are 945 associated events for producer P and 879 associated events for consumer C out of a total of 1817 threat events when considering both the CIRCLE OSINT and Botvrij.eu Data feeds combined. This increased association provides a greater opportunity for correlations between threat events that are shared between both parties.

5.2.2. Benchmark scenario (3)

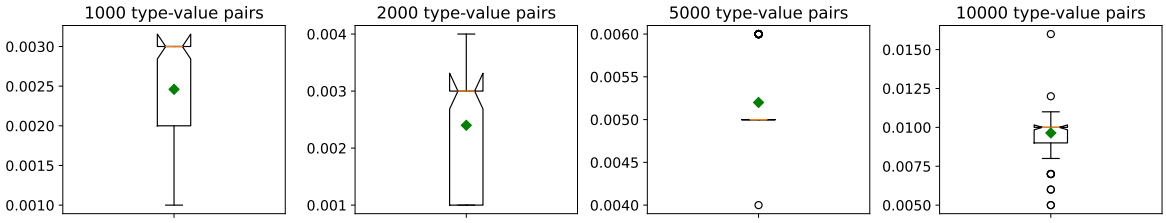
Figure 6 illustrates the performance benchmark results for scenario (3) across four configurations of experiment 1. These configurations involve 1000 threat events and varying numbers of confidential type-value pairs (1000, 2000, 5000, or 10000). As anticipated, as the number of confidential attributes increases from 1000 to 10000 for both producer P and consumer C, the computation time for PSI (step 1) also increases. For instance, the time required for 1000 confidential attributes is approximately 0.65 seconds, whereas it rises to about 6.77 seconds for 10000 confidential attributes. On the other hand, the time needed to compute PGI (step 2) remains below 0.30 seconds. The duration for consumer C to verify all threat events against the Bloom filter created by producer P to identify missing correlations is once again proportional to the number of confidential attributes. It takes less than 0.01 seconds for 1000 confidential attributes but can take up to 7.9 seconds when consumer C has 10000 confidential attributes.

In Figure 7, the results of experiment 2 are depicted when producer P and consumer C randomly select 1500

threat events. The increase in time for the last Bloom filter test from about 7.9 seconds in experiment 1 to about 11.6 seconds for 10000 confidential attributes can be attributed to the additional threat events (i.e., 1000 events versus 1500 events). As expected, the required time increases linearly with the number of events being tested against the Bloom filter. The number of events in experiment 2 has increased by a factor of 1.5, resulting in approximately 1.47 times longer execution time compared to experiment 1. This behavior can be attributed to the characteristics of the Bloom filter used in the experiments. Each element to be checked undergoes a fixed set of hash function calculations, and the corresponding bits in the fixed size bit array are verified for a value of 1. The computational complexity of the hash functions remains consistent, which means that the overall time required for the Bloom filter test is primarily influenced by the number of membership checks performed. As a consequence, the increased number of events in experiment 2 leads to the observed longer execution time. Producer *P* constructed the Bloom filter itself in less than 2 milliseconds. It was configured to store up to 10000 elements with an error rate of 0.001. Serialized to disk, the Bloom filter occupies approximately 18 kilobytes in size.



(a) Producer *P* and consumer *C* each having 1000 random threat events



(b) Producer *P* and consumer *C* each having 1500 random threat events

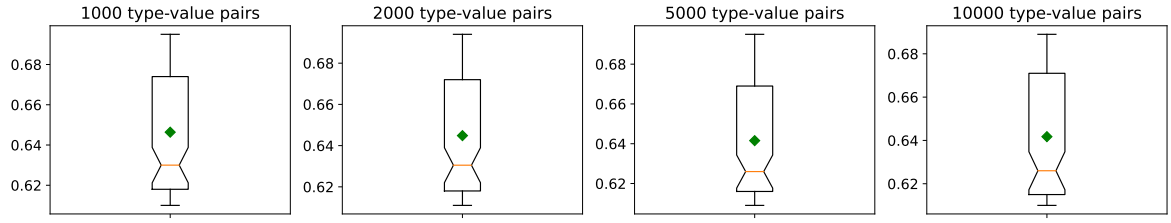
Figure 8: Amount of time spent (in seconds) to compute set intersection via Trusted Third Party (step 1).

From Figures 6 and 7, one can observe that computing the PSI in the first step is a computationally expensive one. To understand the overhead imposed by computing the private set intersection, we compare the time required to compute the common elements via a trusted third party instead. This party hence knows the entire list of producer *P* and consumer *C* but only returns the elements both have in common. The performance impact of this approach is shown in Figure 8. In relative terms, with no consideration for network communication overhead, the computation of the set intersection through a trusted third party takes less than 20 ms, making it two orders of magnitude faster than using PSI. However, it is essential to note that this setup operates under a distinct threat model, assuming the third party behaves honestly and does not engage in malicious activities. Specifically, it is presumed that the third party returns accurate results and does not collude with either producer *P* or consumer *C*.

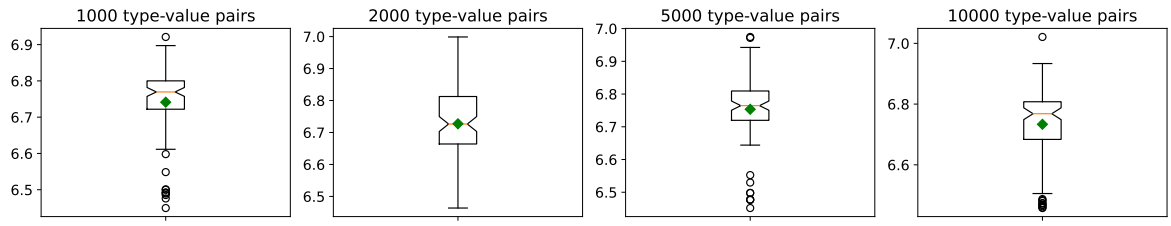
5.2.3. Benchmark scenario (4)

Figure 9 and 10 depict the performance benchmarks for steps 5 and 6 in scenario (4) across all four experiment configurations. We omit the results of step 4 since the analysis is performed locally by MISP's internal correlation engine, as previously explained.

Regarding the results for PSI_3, they indicate that the computation time for the private set intersection of threat event UUIDs between producer *P* and consumer *C* is relatively short. For 1000 threat events in experiment 1, it takes

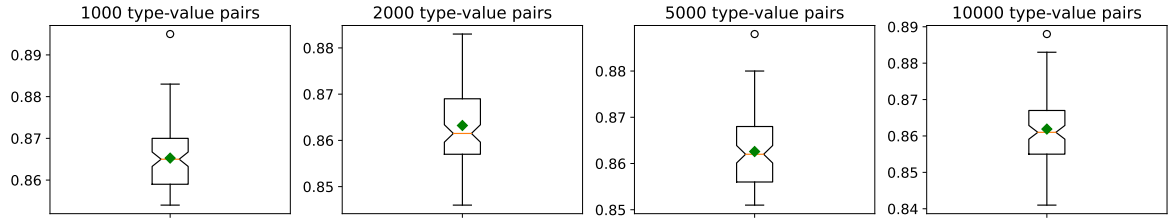


(a) Private Set Intersection 3 (step 5)

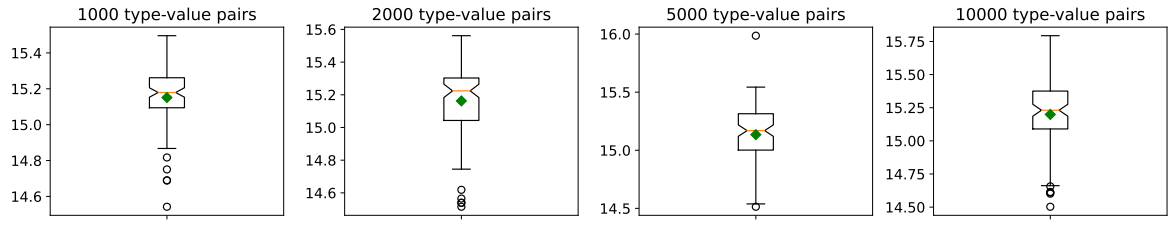


(b) Bloom filter test (step 6)

Figure 9: Amount of time spent (in seconds) for each of the different steps in the algorithms of scenario (4) for producer P and consumer C each having 1000 random threat events.



(a) Private Set Intersection 3 (step 5)



(b) Bloom filter test (step 6)

Figure 10: Amount of time spent (in seconds) for each of the different steps in the algorithms of scenario (4) for producer P and consumer C each having 1500 random threat events.

approximately 0.66 seconds, while for 1500 threat events in experiment 2, it takes around 0.87 seconds. The time required scales linearly with the number of events.

When consumer C performs the Bloom test to detect correlations between each pair of its threat events, it takes about 6.8 seconds for experiment 1 and about 15.5 seconds for experiment 2. As the number of events increases from 1000 to 1500, the time for pairwise testing against the Bloom filter grows quadratically.

The Bloom filter used by producer P had the same parameterization as in scenario (3) and was constructed in less

```
{
  "63ccda3d-9824-4007-818b-bb07b9e66201": {
    "info": "ThreatFox IOCs for 2023-05-19",
    "date": "2023-05-19",
    "analysis": 1,
    "threat_level_id": 2,
    "timestamp": 1684540986,
    "Orgc": {
      "name": "abuse.ch",
      "uuid": "9b086132-8588-49ed-97fd-8578a777822c"
    }
  }

  "c44f4a44-cb96-4e93-bbae-56bf39a20085": {
    "info": "URLhaus IOCs for 2023-05-19",
    "date": "2023-05-19",
    "analysis": 1,
    "threat_level_id": 2,
    "timestamp": 1684540991,
    "Orgc": {
      "name": "abuse.ch",
      "uuid": "9b086132-8588-49ed-97fd-8578a777822c"
    }
  }

  "fd934199-4988-4291-8da3-3093f0566c5e": {
    "info": "MalwareBazaar malware samples for 2023-05-19",
    "date": "2023-05-19",
    "analysis": 1,
    "threat_level_id": 2,
    "timestamp": 1684540981,
    "Orgc": {
      "name": "abuse.ch",
      "uuid": "9b086132-8588-49ed-97fd-8578a777822c"
    }
  }
```


Listing 2: Abuse.ch threat events and IOCs of May 19, 2023

than 2 milliseconds. With the same configuration, the Bloom filter has the same disk size as before.

5.2.4. Discussion on simulated scenarios

The aforementioned experiments are conducted in a simulated environment rather than a real-world one. This approach allows us to compare the impact of an increasing number of threat events and confidential attributes in a systematic and straightforward manner. Additionally, it enables us to validate that our framework enables consumer C to learn correlations between threat events that were explicitly excluded from its dataset.

The primary factor influencing the results is the number of confidential attributes, as each party can have up to 10000 unique type-value pairs, which exceeds what would typically be encountered in real-life scenarios. Moreover, certain confidential attributes may only be temporarily restricted. For instance, specific attributes like source IP addresses can be temporarily privileged to prevent adversaries from realizing that their attack campaigns have been detected. These restrictions can be lifted once intelligence gathering is complete, and appropriate countermeasures have been developed and implemented.

It is worth noting that the PSI, PGI, and Bloom filter benchmarks for scenarios (3) and (4) utilize a single CPU core. Consequently, both experiments can be executed concurrently. Furthermore, by parallelizing each Bloom filter test across all CPU cores, we can further reduce the required time by at least a factor of 3. As a result, the total time required is less than 10 seconds, demonstrating the practical feasibility of the solution.

5.3. Performance impact with abuse.ch intelligence feeds

In our final experiment, we leverage three community-driven threat intelligence feeds from https://abuse.ch, namely ThreatFox, MalwareBazaar, and URLhaus (refer to Table 1). These feeds play a crucial role in combating malware and botnets. Notably, a new MISP event is generated every day across all three feeds. To illustrate, Listing 2 showcases the event UUIDs from these feeds specifically for May 19, 2023. The threat intelligence feeds mentioned above contain various types of event- and object-level attributes, including:

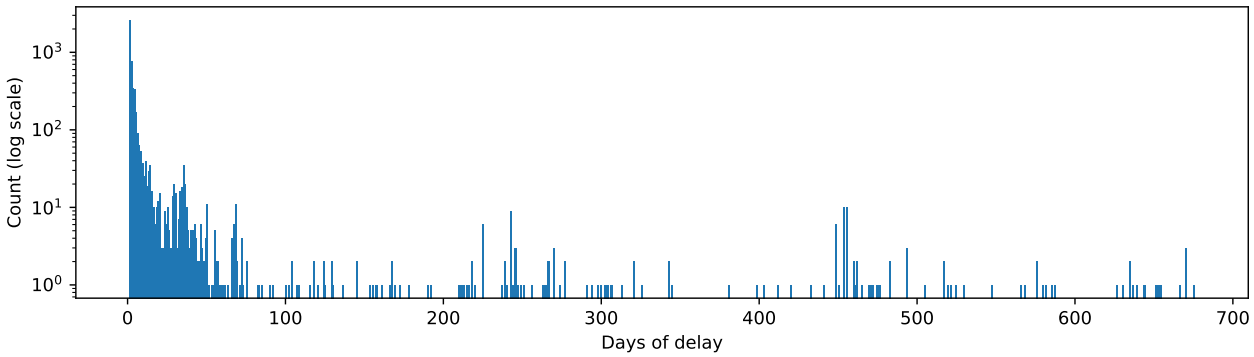


Figure 11: Histogram of maximum delays in days of first occurrence of sha256 entries in the three https://abuse.ch feeds

- **ThreatFox:** domain, ip-dst|port, md5, sha1, sha256, url
- **MalwareBazaar:** filename, imphash, md5, mime-type, sha1, sha256, sha3-384, size-in-bytes, ssdeep, tlsh
- **URLhaus:** domain, ip-dst, url, imphash, md5, mime-type, sha256, size-in-bytes, ssdeep, telfhash, tlsh

Among the mentioned threat intelligence feeds, ThreatFox does not include any object-level attributes, only event-level ones. On the other hand, MalwareBazaar exclusively consists of object-level attributes. URLhaus, however, presents a combination of event- and object-level attributes. All three feeds share a common set of attribute types, such as md5 and sha256 hash values for malware samples. Additional attribute types like sha1 or sha3-384 are only present in certain feeds. It is worth noting that if two samples possess the same sha256 hash value, it is highly likely that the other hash values will exhibit similarities as well.

The unique aspect of these feeds lies in the fact that each day introduces a new threat event, devoid of any inherent semantic meaning. The only potential correlation between different events is the occurrence of the same malware hash on different days. Consequently, we conducted a comparison of sha256 entries across the three feeds. The results indicate that Threatfox had 686624 unique entries (721 duplicates), MalwareBazaar had 353536 unique entries (no duplicates), and URLhaus had 3673734 unique entries (182242 duplicates). The sha256 values in the feeds exhibit a degree of overlap, indicating that there are shared entries among them:

- **ThreatFox** and **MalwareBazaar**: 42196 sha256 entries in common
- **ThreatFox** and **URLhaus**: 162753 sha256 entries in common
- **MalwareBazaar** and **URLhaus**: 104161 sha256 entries in common
- **ThreatFox**, **MalwareBazaar** and **URLhaus**: 15932 sha256 entries in common

Next, we conducted an analysis of the 15932 sha256 entries to investigate whether their initial occurrences were reported on distinct days among the three feeds. Our findings revealed that out of the 15932 sha256 values, 5157 of them were first reported on different days across the feeds. In the majority of cases, the discrepancy in reporting was only a matter of one or two days. However, there were 798 entries that exhibited reporting delays exceeding 7 days, with one particular sha256 entry having a staggering delay of 676 days. Figure 11 presents a comprehensive overview of these findings.

Subsequently, we conducted an experiment where producer P possessed data up until May 19, 2023, while consumer C had data only until April 30, 2023. We then assessed the computational implications of inferring correlations for consumer C in order to determine the ongoing reporting of specific malware samples and the continued activity of certain botnets in May 2023. Our analysis shows that 460010 unique type-value pairs were used in May 2023. For the period before May 2023, there were 19038554 unique type-value pairs. Both sets had 48552 type-value pairs in common.

The PSI computation for both parties' events is completed in under 0.05 seconds. Following this process, producer P can detect the missing days at consumer C. Subsequently, producer P can identify the type-value pairs that occur

during these missing days, which also appeared previously (48552 type-value pairs). A Bloom filter, limited to 100000 entries with an error rate of 0.0001, is then constructed by producer P and shared with consumer C. The construction of the Bloom filter takes less than 1 second. Serialized to disk, the Bloom filter occupies approximately 235 kilobytes in size. When consumer C then evaluates the 19038554 type-value pairs for their presence in the Bloom filter, it takes approximately 241 seconds, which is deemed acceptable given the large number of type-value pairs. It is worth noting that the Bloom filter correctly matched all 48552 entries, indicating no false positives.

6. Conclusion

In this study, we have introduced and assessed a practical polyglot solution for the secure sharing and analysis of confidential or private threat intelligence data. Our approach caters to the distinct requirements of both intelligence feed producers and consumers and is implemented using state-of-the-art platforms.

We have developed a novel method called private graph intersection, which allows for the analysis of correlations among threat events while preserving privacy and accommodating multiple sharing organizations. We have extensively evaluated the security implications and computational overhead of this method, leveraging well-established cryptographic techniques such as private set intersection and Bloom filters. Our analysis − both on simulated and more realistic larger scale scenarios − provides compelling evidence for the practicality and effectiveness of our solution for pivacy-preserving correlation of cross-organizational cyber threat intelligence.

As part of our future research, we will explore methods to incorporate cross-organizational correlation of partial value matches. Additionally, we aim to minimize computational overhead by addressing over-correlation issues that may arise from event and attribute values generating excessively noisy correlations.

Acknowledgements

This research is partially funded by the Research Fund KU Leuven, by the Flemish Research Programme Cybersecurity, and by VLAIO through the CS ICON project "Cyber Security Artificial Intelligence" (CSAI). Work for this paper was supported by the European Commission through the H2020 project CyberSec4Europe (https://www.cybersec4europe.eu/) under grant agreement 830929 and the Horizon Europe project KINAITICS (https://kinaitics.eu/) under grant agreement 101070176.

References

[1] B. Jordan, R. Piazza, T. Darley, STIX Version 2.1. OASIS Standard. 10 June 2021 (2021).
URL https://docs.oasis-open.org/cti/stix/v2.1/stix-v2.1.html

[2] B. Jordan, D. Varner, TAXII Version 2.1. OASIS Standard. 10 June 2021 (2021).
URL https://docs.oasis-open.org/cti/taxii/v2.1/taxii-v2.1.html

[3] T. Darley, I. Kirillov, R. Piazza, D. Beck, Cybox version 2.1.1. part 01: Overview. oasis committee specification draft 01 / public review draft 01. 20 june 2016 (2016).
URL http://docs.oasis-open.org/cti/cybox/v2.1.1/part01-overview/cybox-v2.1.1-part01-overview.html

[4] V. G. Li, M. Dunn, P. Pearce, D. McCoy, G. M. Voelker, S. Savage, K. Levchenko, Reading the tea leaves: A comparative analysis of threat intelligence, in: Proceedings of the 28th USENIX Conference on Security Symposium, SEC'19, USENIX Association, USA, 2019, p. 851–867.

[5] L. Daigle, WHOIS Protocol Specification, RFC 3912 (Sep. 2004). doi:10.17487/RFC3912.

[6] C. Lu, B. Liu, Y. Zhang, Z. Li, F. Zhang, H. Duan, Y. Liu, J. Q. Chen, J. Liang, Z. Zhang, S. Hao, M. Yang, From WHOIS to WHOWAS: A large-scale measurement study of domain registration privacy under the GDPR, in: 28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021, The Internet Society, 2021. doi:10.14722/NDSS.2021.23134.

[7] D. Preuveneers, W. Joosen, TATIS: Trustworthy APIs for Threat Intelligence Sharing with UMA and CP-ABE, in: Foundations and Practice of Security - 12th International Symposium, FPS 2019, Toulouse, France, November 5-7, 2019, Vol. 12056, Springer, 2020. doi:10.1007/978-3-030-45371-8_11.

[8] D. Preuveneers, W. Joosen, J. B. Bernabé, A. F. Skarmeta, Distributed security framework for reliable threat intelligence sharing, Secur. Commun. Networks 2020 (2020) 8833765:1–8833765:15. doi:10.1155/2020/8833765.

[9] D. Preuveneers, W. Joosen, Sharing machine learning models as indicators of compromise for cyber threat intelligence, Journal of Cybersecurity and Privacy 1 (1) (2021) 140–163. doi:10.3390/jcp1010008.

[10] S. Leberknight, Polyglot persistence (2008).
URL http://www.sleberknight.com/blog/sleberkn/entry/polyglot_persistence

[11] C. Wagner, A. Dulaunoy, G. Wagener, A. Iklody, Misp: The design and implementation of a collaborative threat intelligence sharing platform, in: Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security, WISCS '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 49–56. doi:10.1145/2994539.2994542.

[12] D. Kales, C. Rechberger, T. Schneider, M. Senker, C. Weinert, Mobile private contact discovery at scale, in: Proceedings of the 28th USENIX Conference on Security Symposium, SEC'19, USENIX Association, USA, 2019, p. 1447–1464.

[13] D. Preuveneers, W. Joosen, Privacy-preserving polyglot sharing and analysis of confidential cyber threat intelligence, in: Proceedings of the 17th International Conference on Availability, Reliability and Security, ARES '22, Association for Computing Machinery, New York, NY, USA, 2022. doi:10.1145/3538969.3538982.
URL https://doi.org/10.1145/3538969.3538982

[14] A. Zibak, C. Sauerwein, A. Simpson, A success model for cyber threat intelligence management platforms, Computers & Security 111 (2021) 102466. doi:10.1016/j.cose.2021.102466.

[15] X. Bouwman, V. L. Pochat, P. Foremski, T. van Goethem, C. H. Gañán, G. C. M. Moura, S. Tajalizadehkhoob, W. Joosen, M. van Eeten, Helping hands: Measuring the impact of a large threat intelligence sharing community, in: K. R. B. Butler, K. Thomas (Eds.), 31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022, USENIX Association, 2022, pp. 1149–1165.
URL https://www.usenix.org/conference/usenixsecurity22/presentation/bouwman

[16] H. Gascon, B. Grobauer, T. Schreck, L. Rist, D. Arp, K. Rieck, Mining attributed graphs for threat intelligence, in: Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy, CODASPY '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 15–22. doi:10.1145/3029806.3029811.

[17] J. Thom, Y. Shah, S. Sengupta, Correlation of cyber threat intelligence data across global honeypots, in: 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021, pp. 0766–0772. doi:10.1109/CCWC51732.2021.9376038.

[18] G. Gonzalez Granadillo, M. Faiella, I. Medeiros, R. Azevedo, S. G. Zarzosa, ETIP: an enriched threat intelligence platform for improving OSINT correlation, analysis, visualization and sharing capabilities, J. Inf. Secur. Appl. 58 (2021) 102715. doi:10.1016/j.jisa.2020.102715.

[19] C. Martins, I. Medeiros, Generating quality threat intelligence leveraging osint and a cyber threat unified taxonomy, ACM Trans. Priv. Secur. 25 (3) (may 2022). doi:10.1145/3530977.
URL https://doi.org/10.1145/3530977

[20] N. Sun, M. Ding, J. Jiang, W. Xu, X. Mo, Y. Tai, J. Zhang, Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives, IEEE Communications Surveys & Tutorials (2023) 1–1doi:10.1109/COMST.2023.3273282.

[21] A. Weathersby, Prevalence of PII within public malware sandbox samples and implications for privacy and threat intelligence sharing, in: CCSC Eastern Conference 2021, Arlington, VA, USA, 2021.

[22] J. R. Trocoso-Pastoriza, A. Mermoud, R. Bouyé, F. Marino, J.-P. Bossuat, V. Lenders, J.-P. Hubaux, Orchestrating collaborative cybersecurity: A secure framework for distributed privacy-preserving threat intelligence sharing (2022). arXiv:2209.02676.

[23] C. Mouchet, J. Troncoso-Pastoriza, J.-P. Bossuat, J.-P. Hubaux, Multiparty homomorphic encryption from ring-learning-with-errors, Cryptology ePrint Archive, Paper 2020/304, https://eprint.iacr.org/2020/304 (2020). doi:10.2478/popets-2021-0071.
URL https://eprint.iacr.org/2020/304

[24] T. van de Kamp, A. Peter, M. H. Everts, W. Jonker, Private sharing of iocs and sightings, in: Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security, WISCS '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 35–38. doi:10.1145/2994539.2994544.
URL https://doi.org/10.1145/2994539.2994544

[25] S. Dara, S. T. Zargar, V. Muralidhara, Towards privacy preserving threat intelligence, Journal of Information Security and Applications 38 (2018) 28–39. doi:https://doi.org/10.1016/j.jisa.2017.11.006.
URL https://www.sciencedirect.com/science/article/pii/S2214212617300078

[26] R. van Rijswijk-Deij, G. Rijnders, M. Bomhoff, L. Allodi, Privacy-conscious threat intelligence using dnsbloom, in: 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), 2019, pp. 98–106.

[27] J. Freudiger, E. De Cristofaro, A. Brito, Privacy-friendly collaboration for cyber threat mitigation (2014). doi:10.48550/ARXIV.1403.2123.
URL https://arxiv.org/abs/1403.2123

[28] B. H. Bloom, Space/time trade-offs in hash coding with allowable errors, Communications of the ACM 13 (7) (1970) 422–426.

[29] N. Li, T. Li, S. Venkatasubramanian, t-closeness: Privacy beyond k-anonymity and l-diversity, in: 2007 IEEE 23rd international conference on data engineering, IEEE, 2007, pp. 106–115.

[30] C. Dwork, A. Roth, et al., The algorithmic foundations of differential privacy., Found. Trends Theor. Comput. Sci. 9 (3-4) (2014) 211–407.

[31] C. Dong, L. Chen, Z. Wen, When private set intersection meets big data: an efficient and scalable protocol, in: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 2013, pp. 789–800.

Appendix A. Privacy policy for polyglot persistency

```
{
  "version": "1.0.0",
  "creator": "davy.preuveneers@kuleuven.be",
  "organization": "kuleuven",
  "attributes": [{
      "name": "ip-dst",
      "pets": [{
          "scheme": "cpabe",
          "metadata": { "policy": "and foo bar" }
        }, {
          "scheme": "sha256"
        }, {
          "scheme": "pbkdf2",
          "metadata": { "iterations": 1000 }
        }
      ]
    }, {
      "name": "email",
      "pets": [{
        "scheme": "sha256"
      }]
    }
  ],
  "templates": [{
    "attributes": [{
        "name": "pcap_file",
        "type": "IDENTIFYING",
        "pets": [{
          "scheme": "cpabe",
          "metadata": { "policy": "and foo or bar baz" }
        }]
      }, {
        "name": "ip_src",
        "type": "INSENSITIVE"
      }, {
        "name": "ip_dst",
        "type": "QUASI_IDENTIFYING",
        "pets": [{
          "scheme": "pbkdf2",
          "metadata": { "iterations": 1000 }
        }]
      }
    ],
    "name": "custom_network_security_object",
    "pets": [{
      "scheme": "k-anonymity",
      "metadata": { "k": 2 }
    }],
    "uuid": "d2f7910b-f757-4370-9db1-cfa3e89c20b8"
  }]
}
```

Listing 3: Privacy policy for polyglot persistence