Privacy-Preserving Polyglot Sharing and Analysis of Confidential Cyber Threat Intelligence

Davy Preuveneers
imec-DistriNet, KU Leuven
Leuven, Belgium
davy.preuveneers@kuleuven.be

Wouter Joosen
imec-DistriNet, KU Leuven
Leuven, Belgium
wouter.joosen@kuleuven.be

ABSTRACT

Sharing cyber threat intelligence helps organizations analyze and protect against a growing number and sophistication of security threats. However, organizations are reluctant to share their locally collected cyber threat intelligence with third parties because of the the risk of incidentally disclosing sensitive business data or personally identifiable information, and the subsequent reputational harm or even financial repercussions imposed by the GDPR. To address the different confidentiality needs of threat intelligence producers and consumers, we present and evaluate a practical polyglot solution for privacy-preserving sharing and analysis of confidential or private information, and this on top of a contemporary cyber threat intelligence platform. Additionally, we investigate the security impact and computational overhead of these techniques to analyze correlations between threat events in a privacy-preserving manner and across sharing organizations.

CCS CONCEPTS

• **Information systems** → **Information systems applications**; • **Security and privacy** → **Privacy-preserving protocols**; **Access control**.

KEYWORDS

threat intelligence sharing; security; privacy; polyglot persistence and analysis

ACM Reference Format:
Davy Preuveneers and Wouter Joosen. 2022. Privacy-Preserving Polyglot Sharing and Analysis of Confidential Cyber Threat Intelligence. In *The 17th International Conference on Availability, Reliability and Security (ARES 2022), August 23–26, 2022, Vienna, Austria.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3538969.3538982

1 INTRODUCTION

Threat intelligence platforms (TIP) are widely adopted in organizations, enterprises and CERT communities to analyze security incidents and elicit tactical insights to mitigate cyber attacks. Due to the growing number and sophistication of security threats and attack campaigns, an organization can no longer collect and analyze information about security incidents all on its own to effectively and proactively fend off attacks. To enhance situational awareness in a collaborative manner, several data exchange formats – such as STIX [9], TAXII [10] and CyBOX [3] – have been proposed and adopted by TIPs to facilitate the sharing of indicators of compromise (IoC). As a result, organizations can now more easily consume a variety of intelligence feeds – like CIRCLE OSINT[1], Botvrij.eu[2] or the Feodo IP Blocklist[3] – as a source of information to enrich threat intelligence they collected on their own systems and networks. While collecting and sharing security incident data are important functionalities of a TIP, the essential part is weeding through ever larger intelligence feeds [14] of data and filter the actionable information that is relevant to protect the organization and its systems and networks.

While the technical capabilities are available to facilitate information sharing, organizations are reluctant to share their own threat intelligence – such as a post-mortem analysis of an incident and digital forensics gathered after a system compromise – with others, not only because of the risk of reputational harm when their customers learn about the breach, but also due to the risk of publicly disclosing sensitive or private information. The General Data Protection Regulation (GDPR) prohibits publishing personally identifiable information (PII) and may impose severe fines when doing so. For example, WHOIS [1] is a well-known service for finding information on any domain name or website – and hence often used by security analysts – that was impacted by the GDPR. To avoid non-compliance, the service now redacts or anonymizes various fields of the registrant, such as the name, address, email address and phone number. Hence, whenever a domain name is used for malicious purposes (e.g. as a command and control server in a malware campaign), these attributes are no longer available [15] to security analysts to pivot between threats and identify the adversaries behind the attack campaign. Even an IP address – an attribute often used as part of an IoC – is considered as personal data by the GDPR if it relates to an identified or identifiable 'natural' person[4].

Sharing threat intelligence is key to thwart attacks but confidentiality and privacy concerns hamper voluntary reporting efforts. We address this challenge by proposing a practical solution for TIPs that allow security analysts to better balance the security and privacy trade-off when handling business sensitive, private or personally identifiable information. Our policy-driven solution extends our previous research [16–18] to implement a polyglot framework that combines various privacy enhancing techniques to store, process and share threat intelligence in order to adapt to

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ARES 2022, August 23–26, 2022, Vienna, Austria
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9670-7/22/08…$15.00
https://doi.org/10.1145/3538969.3538982

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

the different security and privacy needs of both threat intelligence producers and consumers. Whereas polyglot persistence [12] typically refers to the use of different technologies to handle varying data storage needs, our polyglot solution follows a similar paradigm for multi-faceted processing and sharing of information. More specifically, the threat intelligence is made available in different variations after selective suppression, encoding, hashing and encryption – including combinations thereof – depending on the sensitivity of the information and the security analytics needs of the authorized target audience. Additionally, we investigate the security impact and computational overhead of these techniques on top of the MISP [22] cyber threat intelligence platform, to analyze in a privacy-preserving manner correlations between threat events belonging to different organizations. These techniques build upon Private Set Intersection (PSI) [11], a privacy-preserving cryptographic technique that allows two parties to compare their sets of data points and compute their intersection without revealing the raw data to the other party. The key contributions of this work can be summarized as follows:

(1) We present a polyglot persistence and analysis framework for threat intelligence that adheres to the different needs of intelligence feed producers and consumers.
(2) We propose a novel method to analyze correlations between threat events in a privacy-preserving manner and across sharing organizations.
(3) We investigate the security impact and computational overhead of various privacy enhancing techniques on top of a contemporary threat intelligence platform.

The remainder of this paper is structured as follows. Section 2 discusses relevant related work. The design and implementation of our polyglot persistence and sharing solution is discussed in Section 3. Section 4 explains the privacy-preserving correlation of threat intelligence, while the security and privacy impact as well as the computational complexity are evaluated in Section 5. We conclude this work in Section 6 summarizing the main insights and outlining topics for further research.

2 RELATED WORK

This section will discuss relevant related work on threat intelligence platforms and their added value, correlation analysis and privacy-aware sharing and processing of threat information.

Zibak et al. [24] recently carried out a survey study to investigate measures that influence the effectiveness of threat intelligence platforms. They analyzed responses from 152 security professionals to get a better understanding of the success factors. Their empirical evaluation indicates that the quality of the information and the perceived trust in the platform are among the most important success factors. Similar comparative work on the importance of cyber threat intelligence was carried out by Li et al. [14]. Our practical contribution supports these priorities by offering a solution to even share actionable but private information in a confidentiality- or privacy-preserving manner using state-of-the-art techniques to further strengthen the trust in the TIP and its intelligence.

Gascon et al. [7] present Mantis, a threat intelligence platform that is capable of information retrieval and correlating threat data collected through different threat intelligence standards. To support security analysts with the identification of similar attack patterns in seemingly unrelated attack campaigns, Mantis implements a type-agnostic similarity algorithm based on attribute graphs. Thom et al. [19] also investigated the effectiveness of correlation of cyber threat intelligence data but now across global honeypots simulating actual Internet facing services. Their goal was to compare attack and traffic patterns to learn more about the tactics being employed by adversaries. Our solution aims to support or implement correlation analysis methods for the same purposes, albeit in a privacy-preserving manner.

Gonzalez Granadillo et al. [8] proposed ETIP, an enriched threat intelligence platform with extended capabilities in terms of import, quality assessment processes, visualization and information sharing in current TIPs. It leverages OSINT data as well as data provided by external sources and an organization's IT infrastructure. These feeds are correlated, evaluated and represented as a threat score. Our solution has the same ambition to offer extended capabilities, but specifically for sharing, analyzing and correlating confidential information in a privacy-preserving manner.

The prevalence of personal identifiable information was investigated by Weathersby [23]. More specifically, he examined public malware sandbox samples and their implications for privacy and threat intelligence sharing. His exploratory observation analysis of 1012 random samples of non-malicious PDF files uploaded to online malware scanners indicated that 72% had more than 1 PII indicator present, such as author names, email addresses, IP addresses and credit card data. The ability to turn PII into actionable cyber threat intelligence is exactly what our contribution aims to achieve.

Specifically in the area of privacy-preserving threat intelligence, van de Kamp et al. [20] investigated cryptographic schemes for the private sharing of IoCs and the reporting of sightings. Dara et al. [2] applied homomorphic encryption and private information retrieval to safeguard the privacy of a user querying public threat intelligence services and databases, whereas van Rijswijk-Deij et al. [21] explored Bloom filters as a privacy-enhancing technology to store DNS requests in a privacy-conscious threat intelligence context. Freudiger et al. [6] investigated the practical feasibility of PSI for predictive IP address blacklisting. While we use similar techniques, we believe we are the first to use Private Graph Intersection (PGI) in the context of privacy-preserving sharing and correlation of threat intelligence across organizations.

3 POLYGLOT PERSISTENCE AND SHARING

This work builds upon our previous research TATIS [16, 18] framework atop of the MISP[5], The Hive and Cortex[6] threat intelligence platforms. In the following sections, we will explain how we adapted this framework to address the different confidentiality needs of threat intelligence producers and consumers for sharing and correlating threat events.

3.1 Stakeholder concerns and goals for sharing confidential threat intelligence

Consider the simplified MISP threat event example in Listing 1, depicting an IP address as one of the confidential attributes that

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

```
{
  "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

we aim to protect. When sharing threat intelligence information, producers and consumers are faced with different needs.

Req.

- **Producers**: These stakeholders can opt (1) to not share certain information at all, (2) share the fully detailed threat intelligence with a restricted set of consumers, (3) share the information in such a way that it becomes less sensitive or revealing but still useful for security analysis, or (4) a combination of (2) and (3) when targeting different audiences.
- **Consumers**: These stakeholders process the received threat intelligence (1) to gain more insights into a specific incident and/or on how to effectively respond, (2) to confirm their occurrence or sightings, (3) to enhance the analysis of an existing threat event, and (4) to correlate the incident with other (locally observed) threat events.

To strengthen trust in the ecosystem, the above requirements go beyond the Traffic Light Protocol (TLP) tagging scheme[7] for sharing threat intelligence with appropriate audiences. It obviates the need for a more flexible threat intelligence persistence and processing layer, as well as cryptographic means to enforce access constraints.

3.2 Polyglot persistence of confidential attributes

Contrary to the example in Listing 1, a MISP threat event typically has many more attributes or even encapsulates several MISP objects annotated with multiple attributes. Many of these attributes may be confidential or private and our framework can selectively filter, transform and/or encrypt these attributes. The underlying relational database is strongly typed, so the type of the attribute is changed on the fly to store a larger base64 encoded payload (plaintext or ciphertext) rather than the original IPv4 address. The advantage of leveraging MISP's persistence layer is that no additional functionality is needed to share (encrypted or privatized) threat intelligence with other MISP instances via its push or pull synchronization mechanism.

Initially, TATIS [16, 18] only offered fine-grained access control to threat intelligence information by protecting threat events and attributes with Ciphertext-Policy Attribute-Based Encryption (CP-ABE). For the example event in Listing 1, the original IP address attribute is AES encrypted, and the AES secret key is protected with CP-ABE. The rationale behind this decision is that an AES secret key is typically shorter than most attribute values, enabling a faster CP-ABE decryption of the AES secret key, which can then be used to decrypt the actual threat intelligence with AES. The drawback of AES encryption as a privacy enhancing technique is that it does not allow for analyzing correlations between events and attributes unless the consumer is able to decrypt the protected information, i.e. the consumer has an CP-ABE decryption key constructed with those user profile attributes that match the conditions of the encryption policy with which the AES secret key is encrypted.

Hence, we extended our solution with polyglot capabilities by allowing to store the same attribute in multiple variations as a way to restrict access to sensitive attributes at different levels of granularity, hereby supporting correlation analysis for those threat intelligence consumers without decryption keys:

- **Plaintext**: The threat event and associated attributes are stored in the clear. This variation is intended for insensitive or public information for which no access restrictions apply (e.g. TLP:WHITE threat intelligence).
- **Suppression**: An attribute with a particular type or having a content within a restricted set of values is removed if the information is considered too business sensitive, private, or involving personally identifiable information.
- **Transformed**: The original attribute is not made publicly available, but only after being transformed in one or more ways to match the different needs of the threat intelligence consumers without revealing sensitive details.

With respect to the last category, sensitive threat attributes can be transformed in different ways that constrain the amount of information revealed and who can access the information:

- **Hierarchical encoding**: The value of an attribute is transformed and generalized into a pre-defined hierarchy to reduce the uniqueness of the value. Correlation with other events and attributes will be less exact, but the transformed attribute is still semantically interpretable.
- **Hashed**: The original attribute is replaced with its hashed counterpart. Different schemes are supported, including 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, etc. Attribute correlation is still feasible, but interpretation of the content is sacrificed.
- **Encrypted**: Individual attributes can be encrypted with CP-ABE – each with a different encryption policy – so that only owners of a matching decryption key can retrieve the AES secret key and use the latter to obtain the original plaintext. Attribute correlation and interpretation is limited to a restricted and authorized audience.

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

- **Hybrid**: The hidden values of a set of attributes are used in a key derivation scheme (e.g. PBKDF2) to construct a secret key with which an attribute is protected. This way, only threat consumers that know the hidden values can compute the derived key to decrypt the encrypted attribute.

The above techniques can be applied at the level of attributes, and can be combined. For example, the same attribute information can be provided both in (a) encrypted form (i.e. support in-depth analysis for those with a decryption key) and in (b) hashed form for correlation analysis with other events (i.e. support threat event consumers without decryption key).

Whether these techniques are appropriate from a security or privacy point of view depends on the attribute type, and how they are combined for polyglot persistence. If an attribute can only have a limited set of values, then a pre-computation of all (unsalted) SHA-256 hashes becomes practically feasible, rendering the CP-ABE encryption of the same attribute pointless. Similarly, storing an attribute both in plaintext and encrypted is futile too.

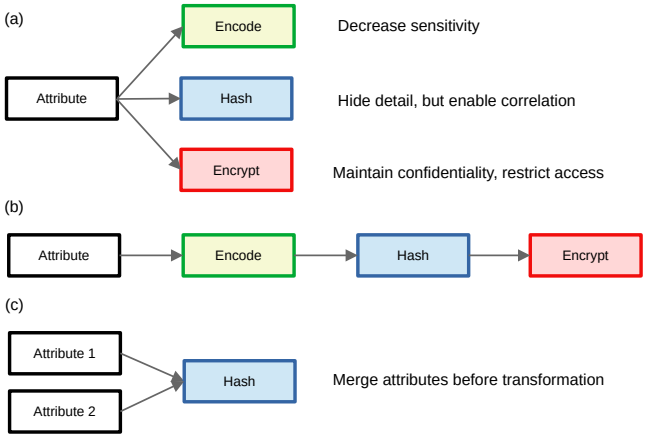

Figure 1: A variety of attribute transformation pipelines

3.3 Composition of transformation techniques

The above transformation techniques can be pipelined or combined in various ways, with some scenarios depicted in Figure 1.

In scenario (a), the original attribute value is not shared in the clear, but rather protected in three different manners, each meeting different purposes for different audiences of threat intelligence consumers as explained in the previous section.

An example for scenario (b): in the first step individual continuous valued attributes are first generalized in an interval-based hierarchy and then top- or bottom-coded. In the second step, the top- or bottom-encoding is transformed with HMAC-SHA-256. This way the structure of the encoding hierarchy itself is hidden, while attributes within the same interval can still be correlated. In a third step, the hashed attributes are encrypted with CP-ABE to restrict access to consumers authorized according to the encryption policy.

Scenario (c) is beneficial when pre-computing the (unsalted) hashes for all possible values of a single attribute type is straightforward, which would allow learning the original attribute value by matching the hashes. For joint attribute values (e.g. combining the IP address and the network port), creating such a lookup table of hashes may not be as practically feasible, but the hash of the joint attributes is still helpful to analyse certain correlations.

Whereas the above methods operate at the level of attributes, our framework also supports the anonymization of threat intelligence at the level of events, not only to mitigate the disclosure of sensitive attributes but also to mitigate membership inference attacks. Currently supported are the following well-known privacy enhancing techniques: k-anonymity, l-diversity, t-closeness [13] and differential privacy [5]. As threat events vary in terms of type and number of attributes, the above methods require a list of attribute types (i.e. the quasi identifiers) that need to be considered for possible identification and anonymization. For example, to achieve k-anonymity the IP address `1.2.3.4` in Listing 1 may be hierarchically encoded into `1.2.*.*`.

Note that the above anonymization techniques are frequently used to release datasets to the community as a whole, whereas in this ecosystem of threat intelligence sharing, threat events are processed and shared individually and incrementally.

3.4 Policy-driven polyglot persistence

The configuration of polyglot persistence of threat intelligence is policy-driven, with an example of such a policy shown in Listing 2 in the Appendix. This policy is set by the security administrator to indicate how sensitive, private or confidential threat intelligence is shared with third parties. It allows for flexible composition of privacy enhancing techniques, as indicated in Figure 1. Furthermore, the polyglot solution makes no distinction between a security analyst creating a new MISP threat event or adding an attribute to enrich an already existing MISP threat event. As such, it supports the full lifecyle of threat intelligence.

With this particular policy in place, we process the attributes of type `ip-dst` and `email`, two attributes that belong to the default attributes and categories that MISP provides out-of-the-box[8]. The same process is executed for attributes of a MISP object instantiated according to a MISP object template[9]. In this case, the policy illustrates (a) how individual attributes are transformed with one or more privacy-enhancing techniques (PETs), and (b) how for each MISP object instantiated according to the object template `custom_network_security_object` the method k-anonymity is applied before the attribute transformations and before the threat event is stored in MISP's relational database and possibly shared with MISP instances of other organizations.

Figure 2 depicts the practical realization of the framework from the point of view of MISP's dashboard. The figure illustrates a mock threat event with two attributes, namely the attribute with type `ip-src` that is stored and shared in the clear, and a second attribute `ip-dst` that is protected according to the policy in Listing 2. For the latter case, the original attribute is disclosed in 3 different ways, including CP-ABE encryption and hashing with SHA-256 and PBKDF2. The 3 different versions of the attribute are bundled in a ZIP file of which the list of files is also shown in Figure 2 as well as the contents of the SHA-256 variant of the `ip-dst` attribute.

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

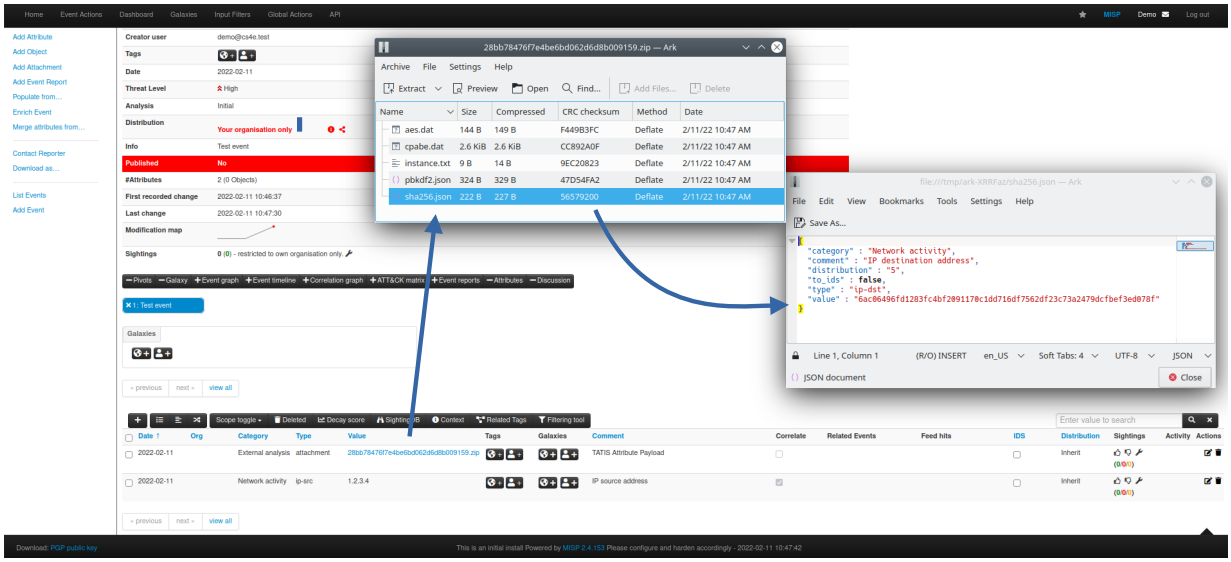

Figure 2: A MISP threat event with (a) a plaintext attribute `ip-src` and (b) an attribute `ip-dst` stored as a ZIP file containing the CP-ABE, SHA-256 and PBKDF2 transformations

Rather than embedding all polyglot variants of an attribute in a ZIP file, our solution can also individually persist the variants in MISP, using the '*Comment*' field to maintain metadata of the polyglot variant, allowing for custom distribution levels per variant.

4 PRIVACY-PRESERVING CORRELATION VIA PRIVATE GRAPH INTERSECTIONS

The TIP assists the security analyst with obtaining actionable information on adversaries by characterizing their attack campaigns and establishing detection patterns to successfully fend off cyber attacks in a timely and possibly fully automated manner. TIPs typically achieve this by comparing threat events of internal and third party intelligence feeds, and finding correlations between these events and the attributes they may have in common. For example, two threat events associated with attacks against different targets may have an embedded attribute in common that designates the source IP address of the two attacks, indicating that the same adversary may be behind the attack campaign.

Our framework supports the establishment of a correlation graph of threat events and attributes across organizations in a privacy-preserving manner. It targets sensitive or confidential attributes that are owned by different organizations and whose contents are not shared with other threat consumers (e.g. TLP:RED threat intelligence), neither in its original nor in a derived form (e.g. encoded, hashed, encrypted). In this scenario, the organizations are willing to help one another with establishing missing links between their own threat events and incidents but without leaking the sensitive threat intelligence at the root of this missing link.

4.1 Correlation graphs of threat events

Correlation graphs in MISP help indicate which threat events are related and what attributes they have in common, such as an IP address of an attacker or a domain name exploited in a phishing campaign. These graphs are instrumental for security analysts to identify clusters of activities and attack campaigns, and to pivot from one threat event to the next.

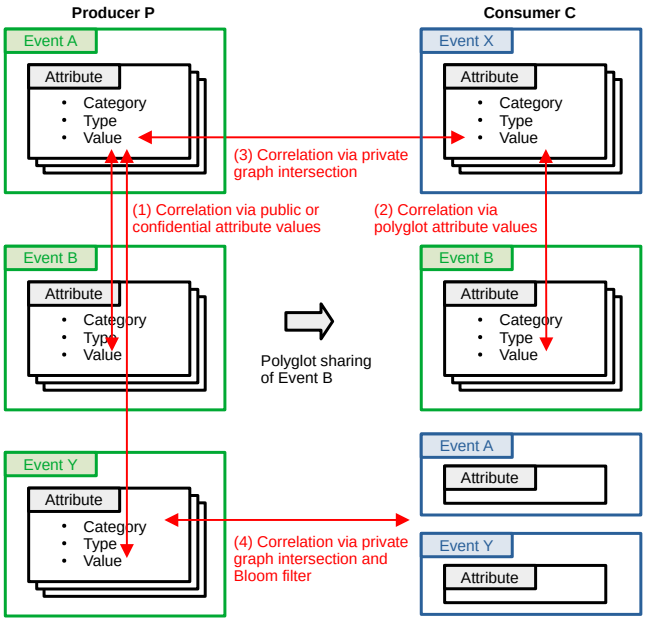

Figure 3: Intra- and cross-organizational correlation of threat events with confidential or sensitive attribute values

Correlations happen at the level of attributes embedded within different threat events, as depicted in Figure 3. These correlations can be induced by either the original contents or by the polyglot variants of the attributes, as discussed in Section 3.2 and depicted in Figure 1:

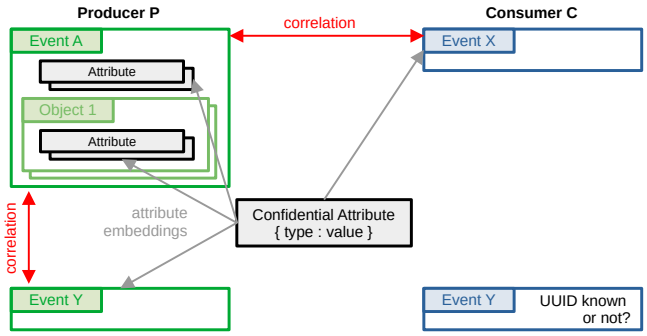

Figure 4: Embedding of confidential attributes in threat events and their objects, inducing correlations between them

- 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 figure depicts two organizations, a threat intelligence producer P and a threat intelligence consumer C. The former knows about three threat events A, B, Y, while the latter knows about threat events A, X, Y. A polyglot variant of Event B is shared with consumer C. Events A and Y are known by both parties, but producer P has additional confidential attributes for these events that are not shared and unknown to consumer C. As such, with non-shared attributes, we refer to those attributes whose values have not been shared in any shape or form between parties. However, producer P and consumer C may have confidential attributes with the same types and values in common, not through sharing but through independent threat intelligence gathering. These confidential attributes may be embedded in different threat events, as depicted in Figure 4.

Figure 3 depicts 4 scenarios (cfr. red text with (n), n ∈ [1..4]). Scenario (1) is the default mode of MISP for locally correlating fully detailed attribute values embedded in the threat events A and B as well as A and Y, including those that are pulled or pushed from other MISP instances. Scenario (2) leverages the same correlation capabilities of MISP between events X and B, but now against the polyglot variant of the shared (and possibly privatized) attributes of event B. Scenario (3) aims to correlate non-shared attributes of two threat events A and X across producer P and consumer C. Scenario (4) aims for a threat consumer C to learn a correlation between two threat events A and Y, where both events are known at both sides, but only the threat producer P was able to induce the correlation due to non-shared attributes unknown to threat consumer C.

Scenarios (1) and (2) can leverage existing functionalities for locally matching (transformed) attributes and constructing the correlation graph. Scenarios (3) and (4) will leverage a private graph intersection (PGI) protocol and Bloom filters to learn about correlated events without leaking sensitive or confidential attribute values unknown to the other party. The last two scenarios will be further elaborated upon in the next subsection.

Algorithm 1 Hash attributes based on their type and value

```
1: procedure HashAttr(attributes)
2:     l ← List()
3:     for a ∈ attributes do
4:         h ← hash(a.type : a.value)          ▷ 128-bit hash
5:         l.add(h)
6:     end for
7:     return l
8: end procedure
```

Algorithm 2 PSI of hashed confidential attributes

```
1: procedure PSIConfAttr(p, c)                 ▷ producer and consumer
2:     s1 ← HashAttr(p.get_confidential_attributes())
3:     s2 ← HashAttr(c.get_confidential_attributes())
4:     s ← s1 ∩ s2                             ▷ private set intersection [11]
5:     return s
6: end procedure
```

4.2 Correlating non-shared confidential attributes with private graph intersection

For scenarios (3) and (4), we will analyze the correlation graphs at the threat intelligence producer P and the threat intelligence consumer C. Both parties hold a private graph of events with non-shared confidential attributes that they do not share with other parties. We will refer to these graphs as $G_P = (V_P, E_P)$ and $G_C = (V_C, E_C)$, where V and E are respectively the list of vertices and the list of edges of the graphs. The vertices denote the threat events with confidential attributes, and the edges indicate correlations between threat events.

Consumer C learns about unknown threat event correlations in two cases: (scenario 3) consumer C and producer P have a confidential attribute in common within threat events $v_{C,i} \sim v_{P,i'}$, and (scenario 4) producer P has two threat events $v_{P,i} \sim v_{P,j}$ in V_P with a confidential attribute in common (scenario 4) with consumer C also aware of these two threat events $v_{C,i'}, v_{C,j'}$ but not of their correlation induced by this confidential attribute.

4.2.1 Scenario 3: Cross-organizational correlation. (Step 1) Producer P and consumer C compute the private set intersection PSI_1 [4] of the non-shared confidential attributes, or more specifically of the 128-bit hash of the type and value fields of these attributes, as illustrated in Algorithms 1 and 2. The rationale for using the hash is that the size of an attribute value can vary from a few bytes (e.g. an IP address) to several megabytes (e.g. a malware sample). Algorithm 2 is a simplified representation of the actual PSI algorithm. Our framework leverages the Low Multiplicative Complexity (LowMC) PSI implementation by Kales et al. [11] [10] due to the performance benefits of LowMC. Producer P and consumer C can reconstruct the original confidential attribute contents from the hashes in the PSI.

(Step 2) Producer P and consumer C each construct their correlation graph, respectively G_P and G_C, for the threat events induced by the confidential attributes in the PSI_1 found in step 1. Next, they

[10] https://github.com/contact-discovery/mobile_psi_cpp

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
```

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
```

Für jedes Attribut in PSI1 check events mit dem Attribut und compute hash

compute the private graph intersection (PGI) of the event correlation graphs G_P, G_C. The intersection of both graphs is defined as $G_I = (V_I, E_I) = G_P \cap G_C$ with $V_I = V_P \cap V_C$ and $E_I = E_P \cap E_C$. The correlations can be represented by 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 edge $e_{i,j}$ is 1 if threat events V_i and V_j have an attribute in common that was part of the PSI_1 of step 1, otherwise 0. After computing the PGI, both parties know which threat event correlations they have in common. Note that the $m \times m$ size of this edge matrix $E_{m,m}$ may be different for both parties. Furthermore, producer P and consumer C may have a different set of threat events. Last but not least, the edge matrix E is typically shallow (i.e. few 1's and many 0's). Hence, both parties encode each correlation (i.e. an edge with value $e_{i,j} = 1$) between threat events V_i and V_j using the 128-bit hash of their UUIDs (i.e. they compute $hash(V_i.uuid : V_j.uuid)$) into a set of hashed event correlations. As correlations are symmetric, V_i and V_j are sorted according to their UUID before computing the hash. Producer P and consumer C then compute the PSI_2 of both sets in a similar manner as for the confidential attributes in step 1 to compute the common correlations or edges E_I of the PGI.

(Step 3) The producer P then selects the confidential attributes that are responsible for threat event correlations not found in the PGI – in fact PSI_2 – found in step 2 and that it is willing to share. It constructs a Bloom filter as depicted in Algorithm 3 to store the embeddings of these attributes within their respective threat events. After receiving the Bloom filter, consumer C can iterate through all the confidential attributes in the PSI_1 of step 1, and for each attribute iterate through the UUIDs of its own threat events to check whether their *hash(a.type:a.value:e.uuid)* is stored in the Bloom filter. If a match is found, then a new correlation has been learned with a threat event that does not yet have this confidential attribute.

Note that a Bloom filter may result in false positives, but not in false negatives. The false positive rate ϵ depends on the number of bits m of the Bloom filter, the number of elements n to store, and the number 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. Producer P identified a correlation between two confidential attributes with the same type and value, and embedded in two different threat events $v_{P,i} \sim v_{P,j}$ in V_P. Consumer C knows about the threat events $v_{C,i'}, v_{C,j'}$ in V_C, but does not have the confidential attribute embedded in them, and as such, no knowledge about the correlation between the two events. Producer P is only willing to share the correlation with consumer C at the level of the threat events – i.e. without revealing the contents of the confidential attributes – if and only if consumer C knows about the threat events in the first place.

gleiche events?

(Step 4) When the confidential attribute is part of the PSI_1 in step 1 of scenario 3, then there is another threat event $v_{C,k'}$ in V_C that, as part of scenario 3, will let consumer C learn about the correlation between $v_{C,k'} \sim v_{C,i'}$ and $v_{C,k'} \sim v_{C,j'}$. Based on this information, it can learn itself about the correlation between $v_{C,i'} \sim v_{C,j'}$.

C kennt Attribut

(Step 5) When the confidential attribute is not part of the PSI_1 in step 1 of scenario 3, and as such, only known to producer P, then consumer C will not learn about the correlation through the protocols of scenario 3. In this case, both parties compute the PSI_3 of the UUIDs of all their non-shared events (the UUID for shared events would already be known if sharing is symmetric). Producer P again computes a Bloom filter for correlated threat events where the correlation is induced by non-shared confidential attributes not in the PSI_1 of step 1 for those common threat events in PSI_3, as illustrated in Algorithm 4.

(Step 6) Consumer C learns about unknown correlations by computing the pairwise hashes $hash(e_1.uuid : e_2.uuid)$ for the event UUIDs in PSI_3 it has in common with producer P, and checking whether they are in the Bloom filter.

5 EVALUATION

In this section, we will specifically evaluate the security and performance impact of correlating non-shared confidential attributes

PSI2 kann nur Intersections haben wenn uuid_p und uuid_c gleich sind wegen hash(V_uuid_i:V_uuid_j)

Kann also unbekanntes Attribut zu Event zuordnen, wenn uuid_p = uuid_c

Wenn ATtribut in anderem Event vorkam

d.h Consumer kennt attribut und event aber nicht dass zwei Event skorrelieren obwohl beide gleiche Attribute haben??

with the private graph intersections and Bloom filters outlined 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 computing the PSI on the confidential attribute hashes, the embedding of the confidential attributes within their respective threat events is not revealed. In other words, one party does not learn the relationship between the confidential attributes and the threat events in which they are embedded at the other party.
- *Step 2:* After the protocol has ended, both parties can learn which threat event correlations they have in common and which of their own threat event correlations are not known by the other party. However, the confidential attributes that induced these correlations at the other party are not revealed. Both parties do not know about the other's threat events except those in the PGI. Last but not least, an attribute with the same type and value, but not marked confidential, is not part of the PGI, and as such does not leak additional information to the other party.
- *Step 3:* The Bloom filter ensures that consumer C does not learn the embeddings of confidential attributes for threat events it does not know about. It would need to guess the UUID, a random 128-bit value. Even if lucky with random guessing, this identifier does not leak any other information about the unknown threat event. Furthermore, a brute force attack would cause mismatches, as a Bloom filter may result in false positives.
- *Step 4:* Consumer C learns about additional correlations through the reflexive and transitive property of correlations. This step has no additional security implications.
- *Step 5:* Both parties learn the UUIDs of the threat events they have in common, but nothing more.
- *Step 6:* Consumer C learns about new correlations between threat events it already knew, but nothing about the confidential attribute of producer P that induced the correlation, nor anything about correlations with another threat event unknown to consumer C.

The privacy-preserving analysis of correlations between threat events heavily builds upon the security properties of PSI [11] and the probabilistic features of Bloom filters. By computing PSI_1 of the confidential attributes, the link with the corresponding threat events is not revealed. Vice versa, PSI_2 to compute the private graph intersection, does not leak the relationship with the confidential attributes. Similarly, PSI_3 of the UUIDs of the threat events does not leak any additional information about the threat event itself. The subsequent correlations are found by the consumer C through querying a Bloom filter. Brute force querying a Bloom filter is computationally intensive and may lead to false positives, a design parameter ϵ that is under the control of the threat intelligence producer P.

A malicious consumer C may aim to exfiltrate information from producer P by triggering a forged PSI using fake but carefully selected confidential attributes. The adversary computes the hashes of fake confidential attributes, and as such may learn the existence of those well-chosen attributes at the producer, but not their respective embeddings in threat events. The adversary may also aim to learn about correlations between threat events, but that would require creating a hash of each pair of all possible UUIDs. Similarly, the probabilistic nature of a Bloom filter prohibits the adversary from brute forcing all combinations of attributes and threat events.

Feed	1	2
1. CIRCLE OSINT Feed	-	1 %
2. The Botvrij.eu Data	49 %	-

Table 1: Threat intelligence feed overlap.

5.2 Performance impact

For the performance evaluation, we make use of the CIRCLE OSINT feed and the Botvrij.eu data. According to the feed overlap analysis matrix of the MISP project [11], there is some overlap between both threat intelligence feeds, as illustrated in Table 1.

Currently, the OSINT feed has 1438 threat events, while the Botvrij.eu feed has 252 events. Each event usually has a set of attributes, and it may have MISP objects that are again characterized by attributes. Considering all kinds of attributes, the former feed has 483215 attributes of which 341402 unique type-value pairs, while the latter feed has 12471 attributes with 12314 unique type-value pairs. 2777 type-value pairs appear in both feeds, i.e. the duplicate attributes. Two threat events with the UUID 5b773e07-e694-458b-b99c-27f30a016219 and 5d9b516c-e5f0-4e7c-a958-5d8c0a019371 appear in both feeds.

5.2.1 Experimental setup. The performance benchmark experiments below represent simulations of rather extreme scenarios in terms of amount of confidential attributes, and hence, computational complexity. These experiments are carried out on a system with an 11th Gen Intel Core i7-11800H CPU running at 2.30GHz and 32GB of memory. After merging both feeds, we configure a MISP instance and our framework for both a threat intelligence producer P and consumer C in two separate virtual machines, each assigned 10GB of memory and 4 virtual CPU cores. Both instances are configured with a set of threat events selected through random subsampling. A first experiment will configure for both producer P and consumer C a random selection of:

- 1000 events
- 1000 attributes marked as confidential

Both parties hence have 1000 attributes marked as confidential, and they may have some confidential attributes in common. A second larger experiment will randomly select for each party:

- 1000 events
- 10000 attributes marked as confidential

Now producer P and consumer C each have 10000 confidential attributes, and now with a higher chance of having the same ones. Also, since both parties have 1000 events, and the merged dataset

[11] https://www.misp-project.org/feeds/

	Events	Common events	Confidential attributes	Common confidential attributes	Confidential attribute embeddings		Associated events	
Experiment 1	1000	599	1000	13	P:842	C:1285	P:182	C:200
Experiment 2	1000	599	10000	581	P:8513	C:9033	P:651	C:639
Experiment 3	1500	1336	1000	11	P:1213	C:1145	P:308	C:307
Experiment 4	1500	1336	10000	567	P:13513	C:14560	P:1027	C:983

Table 2: Producer and consumer threat intelligence statistics.

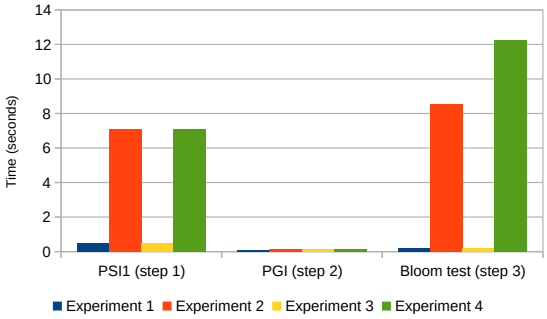

Figure 5: Performance benchmark for scenario 3

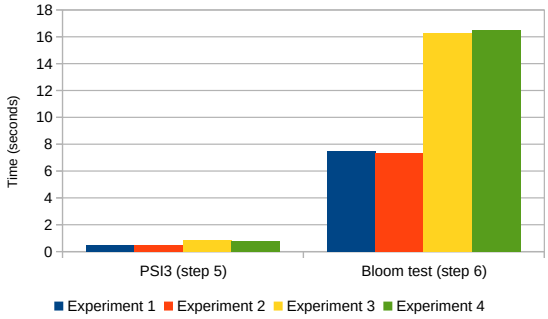

Figure 6: Performance benchmark for scenario 4

only has 1690 events in total (of which 2 with the same UUID), they will have several threat events in common.

We configure a third and fourth experiment in line with the above parameters, but now with producer P and consumer C each having 1500 random threat events. An overview of the intelligence feed statistics for all four experiments is shown in Table 2. For the first and second experiments, the producer P and consumer C have 599 threat events in common (i.e. they share the same UUID), meaning that both of them have 401 unique events. For the first experiment, both parties have 13 confidential attributes in common from the 1000 randomly selected ones. The 1000 confidential attributes of producer P appeared 842 times in its threat events, of which 185 events were unique. Those of consumer C appeared 1285 times in 200 unique threat events. Given that the number of unique threat events is 4 to 14 times less than the number of confidential attribute embeddings, it implies that several confidential attributes are embedded in the same threat events. Similar observations can be made for the second experiment with 10 times as many confidential attributes, and for the third and fourth experiment. Note, however, that for the fourth experiment, producer P and consumer C have a significant larger amount of associated events for their respective confidential attributes (i.e. respectively 1027 and 983 out of 1688 threat events in total for both feeds combined), increasing the opportunity for correlations between threat events that both parties have in common.

To test the privacy-preserving correlation analysis for scenarios (3) and (4), we randomly remove attributes in the 599 threat events that producer P and consumer C have in common. More specifically, we delete 50% of the occurrences of producer P's confidential attributes in consumer C's common threat events as an opportunity for consumer C to learn these now unknown correlations.

5.2.2 Benchmark scenario (3). Figure 5 depicts the performance benchmark results for scenario (3) in each of the 4 experiment configurations. As expected, if the number of confidential attributes for producer P and consumer C increase from 1000 to 10000, so does the time required to compute the PSI (step 1). Whereas for 1000 confidential attributes, the time is about 0.5 seconds, the time increases to about 7 seconds for 10000 confidential attributes. The time necessary to compute the PGI (step 2) is below 0.2 seconds. The time necessary for consumer C to check all its threat events against the Bloom filter created by producer P to learn missing correlations is again proportional to the amount of confidential attributes, with less than 0.25 seconds for 1000 confidential attributes, and up to 12 seconds when consumer C has 10000 confidential attributes. The increase from 8.529 seconds in experiment 2 to 12.256 seconds in experiment 4 can be explained by the additional threat events (i.e. 1000 events versus 1500 events). As expected, the time required increases linearly with the amount of events tested against the Bloom filter.

The Bloom filter itself was constructed by producer P in less than 10 milliseconds. In each experiment, it was configured to store up to 10000 elements with an error rate of 0.001. Serialized to disk, it has a size of about 18 kilobytes.

5.2.3 Benchmark scenario (4). In Figure 6, we illustrate the performance benchmarks for steps 5 and 6 in scenario (4), again for all 4 experiment configurations. The results of step 4 are not shown here as this analysis is carried out locally by MISP's internal correlation engine itself, as explained before. The results for PSI_3 indicate that the time required to compute the private set intersection of the UUIDs of the threat events of producer P and consumer C is relatively small. For 1000 events (experiments 1 and 2), it is about 0.47 seconds, and for 1500 events (experiments 3 and 4) it is around 0.84 seconds. The time required is near linear in terms of the amount of events. The Bloom test by consumer C to check for the presence of correlations between each pair of its threat events takes about 7.4 seconds for experiments 1 and 2, and about 16.4 seconds for

experiment 3 and 4. As the amount of events grows from 1000 to 1500, the time for pairwise testing against the Bloom filter grows quadratic.

The Bloom filter by producer P had the same parameterization as for scenario (3), and was constructed in less than 2 milliseconds for the first 3 experiments and less than 100 milliseconds for the last one. Given the same configuration, the Bloom filter has the same size on disk as before.

5.2.4 Discussion. The above experiments carry out a performance benchmark in a simulated setting rather than a real-world one, not only to easily and systematically compare the impact of a growing number of threat events and confidential attributes, but also to verify that our framework is able to let consumer C learn correlations between threat events that were explicitly removed from its dataset.

The largest impact is due to the amount of confidential attributes, and each party having up to 10000 unique type:value pairs is well beyond of what can be expected in a real-life setting. Furthermore, some of these confidential attributes may only be temporarily restricted. For example, certain attributes, such as source IP addresses, can be temporarily privileged to avoid adversaries from learning that their attack campaigns have been discovered, with restrictions lifted when the intelligence gathering has completed and appropriate countermeasures are developed and put in place.

Last but not least, the PSI, PGI, and Bloom filter benchmarks for scenario (3) and (4) only use a single CPU score, and as such both experiments can be executed in parallel. Additionally, by parallelizing each Bloom filter test over all CPU cores, we can further reduce the time necessary with at least a factor 3. As such, the amount of time required is less than 10 seconds, which demonstrates the practical feasibility of the solution.

6 CONCLUSION

In this work, we presented and evaluated a practical polyglot solution for privacy-preserving sharing and analysis of confidential or private threat intelligence information that adheres to the different needs of intelligence feed producers and consumers and that was built on top of state-of-practice platforms.

We designed a novel private graph intersection method to analyze correlations between threat events in a privacy-preserving manner and across sharing organizations, and investigated the security impact and computational overhead of this method that leverages well-known cryptographic building blocks for private set intersection and Bloom filters to efficiently and effectively share threat intelligence. Our analysis demonstrates the practical feasibility of our solution.

ACKNOWLEDGMENTS

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 No. 830929.

REFERENCES

[1] Leslie Daigle. 2004. WHOIS Protocol Specification. RFC 3912. https://doi.org/10.17487/RFC3912

[2] Sashank Dara, Saman Taghavi Zargar, and VN Muralidhara. 2018. Towards privacy preserving threat intelligence. *Journal of Information Security and Applications* 38 (2018), 28–39. https://doi.org/10.1016/j.jisa.2017.11.006

[3] Trey Darley, Ivan Kirillov, Rich Piazza, and Desiree Beck. 2016. CybOX Version 2.1.1. Part 01: Overview. OASIS Committee Specification Draft 01 / Public Review Draft 01. 20 June 2016. http://docs.oasis-open.org/cti/cybox/v2.1.1/part01-overview/cybox-v2.1.1-part01-overview.html

[4] Changyu Dong, Liqun Chen, and Zikai Wen. 2013. When private set intersection meets big data: an efficient and scalable protocol. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 789–800.

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

[6] Julien Freudiger, Emiliano De Cristofaro, and Alex Brito. 2014. Privacy-Friendly Collaboration for Cyber Threat Mitigation. https://doi.org/10.48550/ARXIV.1403.2123

[7] Hugo Gascon, Bernd Grobauer, Thomas Schreck, Lukas Rist, Daniel Arp, and Konrad Rieck. 2017. Mining Attributed Graphs for Threat Intelligence. In *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy* (Scottsdale, Arizona, USA) *(CODASPY '17)*. Association for Computing Machinery, New York, NY, USA, 15–22. https://doi.org/10.1145/3029806.3029811

[8] Gustavo Gonzalez Granadillo, Mario Faiella, Ibéria Medeiros, Rui Azevedo, and Susana Gonzalez Zarzosa. 2021. ETIP: An Enriched Threat Intelligence Platform for improving OSINT correlation, analysis, visualization and sharing capabilities. *J. Inf. Secur. Appl.* 58 (2021), 102715. https://doi.org/10.1016/j.jisa.2020.102715

[9] Bret Jordan, Rich Piazza, and Trey Darley. 2021. STIX Version 2.1. OASIS Standard. 10 June 2021. https://docs.oasis-open.org/cti/stix/v2.1/stix-v2.1.html

[10] Bret Jordan and Drew Varner. 2021. TAXII Version 2.1. OASIS Standard. 10 June 2021. https://docs.oasis-open.org/cti/taxii/v2.1/taxii-v2.1.html

[11] Danie Kales, Christian Rechberger, Thomas Schneider, Matthias Senker, and Christian Weinert. 2019. Mobile Private Contact Discovery at Scale. In *Proceedings of the 28th USENIX Conference on Security Symposium* (Santa Clara, CA, USA) *(SEC'19)*. USENIX Association, USA, 1447–1464.

[12] Scott Leberknight. 2008. Polyglot persistence. (2008). http://www.sleberknight.com/blog/sleberkn/entry/polyglot_persistence

[13] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd international conference on data engineering*. IEEE, 106–115.

[14] Vector Guo Li, Matthew Dunn, Paul Pearce, Damon McCoy, Geoffrey M. Voelker, Stefan Savage, and Kirill Levchenko. 2019. Reading the Tea Leaves: A Comparative Analysis of Threat Intelligence. In *Proceedings of the 28th USENIX Conference on Security Symposium* (Santa Clara, CA, USA) *(SEC'19)*. USENIX Association, USA, 851–867.

[15] Chaoyi Lu, Baojun Liu, Yiming Zhang, Zhou Li, Fenglu Zhang, Haixin Duan, Ying Liu, Joann Qiongna Chen, Jinjin Liang, Zaifeng Zhang, Shuang Hao, and Min Yang. 2021. 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. https://doi.org/10.14722/NDSS.2021.23134

[16] Davy Preuveneers and Wouter Joosen. 2020. 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. https://doi.org/10.1007/978-3-030-45371-8_11

[17] Davy Preuveneers and Wouter Joosen. 2021. Sharing Machine Learning Models as Indicators of Compromise for Cyber Threat Intelligence. *Journal of Cybersecurity and Privacy* 1, 1 (2021), 140–163. https://doi.org/10.3390/jcp1010008

[18] Davy Preuveneers, Wouter Joosen, Jorge Bernal Bernabé, and Antonio F. Skarmeta. 2020. Distributed Security Framework for Reliable Threat Intelligence Sharing. *Secur. Commun. Networks* 2020 (2020), 8833765:1–8833765:15. https://doi.org/10.1155/2020/8833765

[19] Jay Thom, Yash Shah, and Shamik Sengupta. 2021. Correlation of Cyber Threat Intelligence Data Across Global Honeypots. In *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*. 0766–0772. https://doi.org/10.1109/CCWC51732.2021.9376038

[20] Tim van de Kamp, Andreas Peter, Maarten H. Everts, and Willem Jonker. 2016. Private Sharing of IOCs and Sightings. In *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security* (Vienna, Austria) *(WISCS '16)*. Association for Computing Machinery, New York, NY, USA, 35–38. https://doi.org/10.1145/2994539.2994544

[21] Roland van Rijswijk-Deij, Gijs Rijnders, Matthijs Bomhoff, and Luca Allodi. 2019. Privacy-Conscious Threat Intelligence Using DNSBLoom. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. 98–106.

[22] Cynthia Wagner, Alexandre Dulaunoy, Gérard Wagener, and Andras Iklody. 2016. MISP: The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform. In *Proceedings of the 2016 ACM on Workshop on Information*

Sharing and Collaborative Security (Vienna, Austria) *(WISCS '16)*. Association for Computing Machinery, New York, NY, USA, 49–56. https://doi.org/10.1145/2994539.2994542

[23] Aaron Weathersby. 2021. Prevalence of PII within Public Malware Sandbox Samples and Implications for Privacy and Threat Intelligence Sharing. In *CCSC Eastern Conference 2021*. Arlington, VA, USA.

[24] Adam Zibak, Clemens Sauerwein, and Andrew Simpson. 2021. A success model for cyber threat intelligence management platforms. *Computers & Security* 111 (2021), 102466. https://doi.org/10.1016/j.cose.2021.102466

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 2: Privacy policy for polyglot persistence