PURE: Generating Quality Threat Intelligence by Clustering and Correlating OSINT

Rui Azevedo, Ibéria Medeiros and Alysson Bessani
LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
razevedo@lasige.di.fc.ul.pt, imedeiros@di.fc.ul.pt, anbessani@fc.ul.pt

***Abstract*—Cybersecurity has become a top priority for most organizations. To more aptly protect themselves, organizations are moving from reactive to proactive defensive measures. They are investing in cyber threat intelligence (CTI) to provide them forewarning about the risks they face, as well as to accelerate their response times in the detection of attacks. A mean to obtain CTI is the collection of open source intelligence (OSINT) information via threat intelligence platforms and their representation as indicators of compromise (IoC). However, most of these platforms are providing threat information with little to no processing, presenting thus limitations on generating useful quality data. This work presents an approach for improving OSINT processing to generate *threat intelligence of quality* in the form of *enriched IoCs*. This improved intelligence is obtained by correlating and combining IoCs coming from different OSINT feeds that contain information about the same threat, aggregating them into clusters, and then representing the threat information contained within those clusters in a single *enriched IoC*. The approach was implemented in the PURE platform and evaluated with 34 OSINT feeds, which allowed the creation of enriched IoCs that permitted the identification of attacks not previously possible by analyzing the IoCs individually.**

***Index Terms*—Cybersecurity, threat intelligence (TI), open source intelligence (OSINT), threat intelligence platforms (TIPs), security**

I. Introduction

Cyberattacks have been a constant on the Internet, in such a way that their impact and cost has risen to the billions of dollars. The estimated cost of cybercrime is expected to reach $2.1 trillion in 2019 [1] and more than $6 trillion by 2021 [2]. This value reflects both the relevance of cyberspace in society, as well as the evolution of a threat landscape that moved from teen hackers to increasingly organized teams [3], [4].

The nature of current cyber-criminals requires the implementation of measures able to cope with the novelty of their attacks. This implies a dynamic response which translates in the implementation of one or more technologies, such as advanced malware detection, event anomaly detection, and intelligence-driven defense [3].

Cyber threat intelligence (CTI) – the core of an intelligence-driven defense – has emerged as a critical asset in the fight against cyberattacks, improving organizations defense by facilitating the detection of their activities and allowing the anticipation of future attacks. One way to obtain CTI is by accessing open source intelligence (OSINT) feeds through threat intelligence platforms (TIP). OSINT contain security events about cyberspace threat activities, which are represented as a form of *indicators of compromise* (IoC).

Despite the improvement of TIP's technology, recent studies indicate that limitations are still present [5], in particular, it is hard to harness the volume of threat information produced and shared in the different formats of IoCs. Moreover, most of the TIPs are providing threat information with little to no processing, which makes finding relevant intelligence from them a hard task. This fact has increased the pressure on security analysts, who are already faced with the arduous task of sorting the multitude of alerts generated by their security systems, by forcing them also to sort these additional data flows. Therefore, an increase in the quality of the obtained intelligence is fundamental to make it more useful. This increase should provide context and prioritization, which such intelligence currently lacks [6].

This paper presents an approach for improving OSINT processing to generate *threat intelligence of quality*. This improved intelligence translates into new *enriched IoCs* obtained by correlating and combining IoCs coming from different OSINT feeds that contain information about the same threat. Our approach proposes a novel method for establishing correlations – the *n-level correlation* – that connect different IoCs based on two similarity measures we propose. The method allows the creation of clusters that can be summarized and converted into an enriched IoC, which can permit the discovery of previously unidentified patterns and the detection of new sophisticated attacks.

The paper also presents the *Platform for qUality thReat intelligencE*, PURE, that implements our approach. The platform uses the MISP TIP [20] to collect the OSINT data and store the processed (enriched) IoCs. PURE was evaluated with 34 OSINT feeds, which allowed the creation of enriched IoCs that enabled the identification of cyberattacks not previously possible by analyzing the received IoCs individually.

In summary, the main contributions of the paper are:

1) An approach for improving organizational cybersecurity-based OSINT processing to generate quality threat information in the form of enriched IoCs;
2) A correlation method and two similarity measures to correlate and aggregate IoCs;
3) A platform that implements the approach and an experimental evaluation that shows the ability of this platform to create enriched IoCs.

II. Background and Related Work

A. Key Concepts of Threat Intelligence

The novelty of the agents active in the recent threat landscape forced organizations to rethink their defensive stance, moving from a reactive posture to a proactive one, which requires knowledge, in the form of threat intelligence, to be readily available.

Gartner defines threat intelligence (TI) as evidence-based knowledge, including context, mechanisms, indicators, (...) about an existing or emerging advice, an existing or emerging menace or hazard to assets that can be used to inform decisions regarding the subjects response to that menace or hazard [8]. A shorter definition of TI in the same line is "the process of acquiring, via multiple sources, knowledge about threats to an environment." This knowledge must be actionable, meaning that it must have context, be up to date and must be acted upon; otherwise, it will fail to serve its purpose of protecting the organization [9].

TI is shared as *indicators of compromise* (IoC), i.e., information artifacts that aggregate data on malicious activity in a system or within a network. They present data obtained from forensic analysis of a system or network that was attacked. This TI can be either structured, in a computer-readable format (e.g., STIX) or unstructured, like a free text providing descriptions or assessments of a threat or situation, usually providing context to technical data [10].

The quality of TI is measured along four dimensions [11]:

- *completeness*: how much the information contained in an intelligence artifact allows the identification of an attack;
- *accuracy*: how much the intelligence reduces the number of false positives;
- *relevance*: how much the intelligence relates to the specific purpose for which it is intended;
- *timeliness*: measures the time between the creation of an intelligence artifact and when it reaches its target, either human or defensive infrastructure.

As the previous definitions indicate, TI is the result of the conversion of data into an actionable product that allows a decision. This process is called the threat intelligence life cycle, which is composed of five stages: planning and direction, collection, processing and exploitation, analysis and production, dissemination and integration [12]. This last step was traditionally made through standard communication processes, e.g., meetings, phone calls or emails. More recently the evolution of this field transformed the sharing process with the use of specialized websites, automated distribution feeds and the emergence of TIPs. The latter being specialized tools that should satisfy the following requirements: facilitate the exchange of information; enable automation; facilitate the generation, refinement and vetting of data [13].

Since the definition of the concept of TIP, there has been significant growth in the market of this type of tool with the emergence of both commercial and open source solutions. These solutions typically vary in objectives, capabilities and the scope of their actions. Despite their differences, these tools usually provide some or all of the following capabilities: collection and normalization of information from multiple sources and in multiple formats; correlation and enrichment of data to provide context; categorization into IoCs; integration of derived information into downstream security prevention and detection tools; coordination of the workflow of multiple users during incident response; sharing capabilities with other organizations [14].

B. TIPs Limitations and Enhancing TI Quality

While there are multiple advantages to the use of TIPs, such as reducing the time to react to new threats and/or distributed attacks, their implementation still faces multiple challenges as identified in studies made by Meng et al. [15] and ENISA [5]. In this paper, we focus on three specific TIPs limitations (challenges) with the aim to obtain quality TI in an automated and fast way.

1) Maximizing OSINT sources and reducing the quantity of information that reaches a security analyst: Liao et al. observed that TI is dispersed through multiple locations, time periods, and intelligence artifacts [16]. This dispersal implies that to maximize the knowledge on a specific attack or threat, a security analyst must identify and access multiple sources. Furthermore, ENISA reported that shared TI usually was too voluminous and/or complex to be acted upon [5]. This overflow of information creates multiple interconnected negative impacts, such as hindering the work of the security analysts that have to identify useful intelligence [6]. Both these issues make the analyst's task harder and create the risk of useful information being overlooked.

To address these challenges, we propose to increase the number of sources from which we collect intelligence (improving coverage) and the processing of this information (filtering and aggregation) to be made according to predefined criteria. The end product of our processing will be intelligence represented in a single enriched IoC per threat, reducing the volume of information that security analysts need to inspect.

2) Increasing the quality of intelligence that arrives to an analyst: To guarantee an increase in the quality, the four factors of quality introduced in Section II-A must be addressed. This can only be achieved by working on both the *TIP configuration* and the *TIP internal processing capabilities* [5].

Our approach aims to act on both levels. At the configuration level, we aim at using sources containing high-level TI, i.e., IoCs with high-level information (e.g., vulnerabilities), as opposed to feeds that only offer a low-level TI (e.g., blacklisted IPs). At the internal platform level, we aim at employing filters to extract the relevant information and correlation techniques capable of interconnecting different security events related to the same threat.

3) Facilitate the automation of the generation of improved intelligence: Any proposed platform must be designed having in mind that it should be able to function with minimal human intervention after being started and that it should be possible to integrate its products into other technological platforms.

We proposed an approach that comprises a set of modules that are capable of interacting between them autonomously and automatically in order to generate new quality TI that can be integrated into defense equipment and mechanisms, without requiring any intervention after the initial launch.

III. Quality Threat Intelligence Approach

A. Overview

The approach proposed in this paper involves generating enriched IoCs – *enr*-IoCs, i.e., IoCs that contain high-quality intelligence interconnected. This is done by collecting singular IoCs provided from different OSINT sources, aggregating them in clusters, correlating IoCs within clusters, and then generating new IoCs that represent the most relevant threat information of clusters in a single form. The approach comprises six phases, as shown in Fig. 1:

1) *Collecting TI.* We start by collecting TI from OSINT feeds and other sources, such as TIPs. The feeds and the TIPs are channeled to receptors, which store IoCs temporarily until they are processed. The OSINT sources must be selected adequately in order to provide information relevant to the purpose and with different timings. On the other hand, we can use various TIPs to take advantage of different capacities they have, such as the enrichment of OSINT by resorting to external information that does not come with it (e.g., *asn* command) [17].
2) *Normalization.* Comprises the normalization of the different IoC formats in a single one, since OSINT feeds and TIPs outputs are represented by a wide variety of formats.
3) *Deduplicator.* Compares the IoCs received with the IoCs stored in the database, using a metric of similarity that infers the existence of duplicates, and discards IoCs that provide no new information.
4) *Filtering.* A filtering step is done over the single IoCs for the level of relevance to our purpose to create a threat of intelligence of quality. The set of IoCs of interest resulting from the filter is then sent to the clustering module.
5) *Clustering.* Applies a similarity and weighs metrics over the IoCs of interest to aggregate similar and related IoCs, resulting in clusters that represent similar threats (potentially the same). Also, correlates the IoC attributes within clusters to find the most relevant information that characterizes a threat.
6) *Representing clusters.* Converts the clusters into a single enr-IoC and stores it in a database from which it can later be recovered. Also, the new data can be exported to be used in other systems (e.g., IDS, SIEMs).

B. IoC Similarity

The approach proposed to analyze the IoCs resorts to basic *set theory* This means that an IoC can be conceptualized as a set whose elements are its attributes. The attributes that compose the set are tuples with two fields, $<$ *type of indicator* , *value*$>$. When using this representation, we can observe that given two distinct IoCs, A and B, they can present one of four cases:

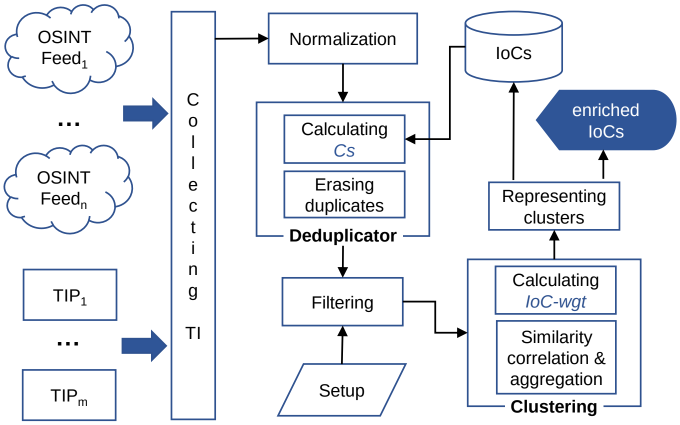

Fig. 1. The quality threat intelligence platform architecture overview.

- *Case 1* - they are completely unrelated ($A \cap B = \emptyset$), in which case we have disjunct sets;
- *Case 2* - one or more elements appear in both sets ($A \cap B \neq \emptyset \wedge A \not\subset B \wedge B \not\subset A$), in which case we consider they are related, indicating that the two IoCs share certain attributes but differ on others, and the union of these sets allows the creation of a new IoC with more information about the same threat;
- *Case 3* - all elements of one set exist in the other ($A \cap B \neq \emptyset \wedge (A \subset B \vee B \subset A)$), in which instance IoC A does not provide any additional information as it is included in IoC B or vice-versa;
- *Case 4* - all elements are equal in both sets ($A = B$), in which case both IoCs are identical.

The Case 2 is the basis of working of the clustering module (see Section IV-B), whereas Cases 3 and 4 are the typical targets for the deduplicator module (see Section IV-A).

To solve these cases, we employ the *Jaccard similarity* index that measures how much two sets are similar, obtained by $J(A, B) = |A \cap B|/|A \cup B|$ [21]. The resulting value is in the range $[0, 1]$. If it takes the value 0 or 1, Cases 1 and 4 are observed, respectively. Otherwise, either Case 2 or 3 can be observed, i.e., the two sets (i.e., IoCs) share elements but are not identical. However, the index does not distinguish between these two cases, and so it is not possible to determine if both IoCs will belong to the set of interest (Case 2) or one of them will be removed (Case 3).

To distinguish both cases and to identify duplicated IoCs, we defined the *contained similarity* indicator, *Cs*, presented as $Cs(A, B) = |A \cap B|/min(|A|, |B|)$. Cs allows the comparison of the cardinality of the intersection of the two sets with the cardinality of the smallest of the sets. When the value of Cs is 1, it means that either one IoC is contained in the other or both IoCs are identical (Cases 3 and 4). On the other hand, when Cs is different from 1, we have Case 2.

IV. Enhancing Threat Intelligence

This section details our method for processing IoCs to generate improved threat intelligence. This method comprises the second, third and fourth phases of our approach – *deduplication*, *filtering* and *clustering*, as presented in Fig. 1.

A. Removing Duplicated IoCs

The deduplication phase aims to eliminate the duplicated IoCs found. It uses the *Cs* indicator to identify duplicated IoCs and discard them. However, in case we find two identical IoCs in terms of attributes, it inspects which IoC contains more valuable data to keep it.

The deduplication phase starts when it receives an IoC resulting from the normalization phase. It checks if the IoC can be related with IoCs already existed in the database, by verifying if there are attributes of this IoC (i.e., the values of their attributes) that are equal to the attributes of IoCs contained in the database. In such a case, it gets those IoCs and creates a set of related IoCs. Next, for each IoC from the set, it creates a pair with the received IoC and then calculates the *Cs* indicator to determine if its value is 1. If so, the removing task is initiated to discover which attributes contain information more relevant and, therefore, to keep this information in the IoC that will be kept. This is determined by the size of information that each attribute carries. However, in situations where two IoCs with identical size are present, the meta-data of each of them will be evaluated, prevailing the one that has a higher trust level or the oldest one (in case they have the same trust level). The trust level represents the level of quality of the IoC, as assigned by a security analyst after he ascertains its information, classifying it as trusted or untrusted.

B. Aggregating and Correlating IoCs

The *clustering* phase comprises the *aggregation* and *correlation* tasks, being both tasks intrinsically connected and laid on the *IoC weight* indicator (*IoC-wgt*, presented next). However, before performing these tasks, an initial *filtering* stage (see Fig. 1) must be performed to identify a *subset of IoCs of interest* in a given context and eliminate IoCs that do not bring added value, such as IP blacklists. Once this subset has been established, the module starts searching for connections between the different IoCs present in the subset and aggregates the related IoCs it finds into clusters. This task is performed based on the calculation of *Cs* indicator, where all pairs of IoCs with $Cs \neq 1$ are considered related, and therefore, selected for aggregation. Next, such IoCs are grouped in clusters, each one representing a single threat. The result of this process is a set of clusters, where each one comprises IoCs that contain relevant interrelated data.

The next stage is processing each cluster to figure out correlations between the IoCs. We defined the *n-level correlation* method to execute this task (see below). This method allows the identification of groups of correlated IoCs within a cluster, resorting of the *Jaccard* index and *IoC-wgt* indicator to discover such groups.

1) IoC weight indicator: When establishing correlations between two related IoCs that contain distinct information, it is useful to determine the enrichment of the resulting *enr*-IoC in order to obtain a measure of the increase in information that was obtained. Also, it is useful to determine how much each IoC contributed to the resulting IoC.

To obtain this, firstly we calculate the *Jaccard* index. However, this measure should be complemented by another indicator as it will not always provide a clear view of the enrichment obtained. For example, given the IoCs A and B, respectively with 1000 and 10 attributes, i.e., A and B have a significant difference in number of attributes, and supposing the majority of the attributes of B belong to the larger set (e.g., 9 out of 10), the index will tend to zero, i.e., $J(A, B) = 0.00899$. In this case, the index does not provide a clear view of the enrichment that will arise from merging the two sets, in part due to the fact that two sets may have completely different sizes. To mitigate this limitation, we propose the *IoC weight* indicator (*IoC-wgt*), presented in Equation 1, which represents quantitatively how much a IoC-pair-relation contributes within of cluster.

$$\textit{IoC-wgt}(A, B) = \frac{\mid A \cap B \mid \times (\mid A \mid + \mid B \mid)}{2\times \mid A \mid \times \mid B \mid} \tag{1}$$

The *IoC-wgt* is obtained from calculating the average of the weight of the shared attributes represented in each IoC of a pair. By using the average of these fractions, we manage to calculate a value that is representative of the extent of different attributes in each of the two IoCs, independently of the relative size of the two IoCs. This means that the value of the *IoC-wgt* provides a measure of the degree of similarity of two sets in relation to each other even when the two sets significantly vary in size, something not possible using only the *Jaccard* index. This allows us to determine the degree to which the union of the two sets will allow an actual increase in the information contained in the resulting set. Returning to the previous example, *IoC-wgt(A,B)* would be 0.4545, which allow us to establish that the contribution of one of the IoCs will not be significant, contrary to what the calculation of the *Jaccard* index for the same pair of IoCs would lead us to expect.

2) The n-level correlation method: The n-level correlation method allows the integration of all IoCs that are interconnected, even if they do not share attributes directly. This method allows the identification of connections previously not identified, being hidden by the lack of completeness of the original IoCs. Liao et al. [16] reported that intelligence may be distributed in multiple sources and only with the observation of all these elements it is possible to construct a complete view of an incident or campaign. Therefore, within the cluster, each IoC will be analyzed with others, however, in this instance, if the IoC shares attributes with other IoCs, then the connections of these IoCs will also be analyzed until all elements in the cluster have been identified and all connections followed. To employ this method, an undirected graph is built by defining its nodes as being the IoCs and the edges the connection of the nodes (IoCs) that share attributes, represented by the pair's *IoC-wgt* indicator. The created graph is then processed to identify sub-graphs of interconnected nodes, extracting thus the most relevant information which forms the *enr*-IoCs.

V. PURE

We implemented our approach in the PURE platform. The platform, developed in Python, is composed of two main modules – *deduplicator* and *enr-IoC generator*. The former identifies and removes duplicated IoCs, whereas the latter aggregates and correlates IoCs in clusters, and then represents such correlated IoCs as new enriched IoCs. In addition, PURE integrates with the MISP [20] platform to perform the role of collector, format normalizer, and storage database. The use of the collector and format normalizer within MISP allowed the design of a solution-oriented to deal with IoCs in MISP format and stored according to the MISP database structure.

In the development of the platform, we made two assumptions: (A1) the level of analysis associated to an event directly correlates to the level of trust that can be placed on the information it contains; (A2) all blacklist events are correctly tagged.

The platform was built by mounting a Docker Container with a MISP instance (version 2.8.69) running on a Dell PowerEdge R420 server with an Intel Xeon E5-2407 CPU 2.2GHz/10 MB cache processor, 32 GB RAM and a 300 GB disk.

PURE has an interface that allows the user to define the filter parameters, which are used to extract the set of IoCs of interest. These parameters are:

- *Trust level*, to ascertain the quality of the IoCs we want to use as a basis for our analysis, and whose value varies from 0 (untrusted, as the IoC has not been analyzed) to 2 (trusted, in which case the IoC has undergone review by a human analyst);
- *Inclusion of blacklists*, to define if IoCs labeled as IP blacklist should be removed (or not) from the set of IoCs that will be processed;
- *Attributes*, allows the selection of which attributes categories and types we wish to consider when analyzing IoCs;
- *Connection selection*, option to select different rules in the definition of which similarities should be taken into consideration, such as accept all relationships, exclude all relationships based on attributes belonging to the MISP 'Network' category or only include relationships based on attributes belonging to the MISP categories 'Attribution', 'Targeting data' or of the type 'vulnerability'.

VI. Experimental Evaluation

The objective of the experimental evaluation was to answer the following questions:

1) Is the platform capable of identifying duplicated IoCs and discarding them?
2) Is the platform capable of aggregating IoCs in clusters, representing them as single malicious threats?

TABLE I
Distribution of OSINT IoCs by organization.

Organization	**Trust level**			**Number of IoCs**	**Avg. Attr. by IoC**
	0	**1**	**2**		
CERT-RLP	0	0	1	1	62
CIRCL	182	138	484	804	126
CUDESO	0	1	121	122	40
CiviCERT	0	0	1	1	79
Crimeware	0	0	1	1	26
CthulhuSPRL.be	7	1	209	217	422
ESET	0	0	1	1	263
FOXIT-CERT	0	0	1	1	180
INCIBE	0	1	0	1	64
NCSC-NL	0	1	1	2	125
clearskysec.com	1	0	0	1	110
inThreat	0	20	2	22	6232
Total	**190**	**162**	**822**	**1174**	**286**

3) Is the platform capable of applying the n-level correlation method to generate *enr*-IoCs?

In order to validate our approach, we evaluated PURE with a set of IoCs collected from diverse and different OSINT feeds. Section VI-A presents the dataset we used in the evaluation, Section VI-B evaluates the deduplicator module and answer the question 1, and Section VI-C assesses the enr-IoC generator module and answers questions 2 and 3.

A. OSINT Dataset Characterization

The MISP platform was configured to collect security events from 34 OSINT feeds provided by 12 distinct organizations, which allowed the collection of 1174 IoCs during 35 days. Table I presents the distribution of events by the organizations (column 5) and the event trust level (columns 2 to 4) based on the analysis indicator present in the IoC. The last column indicates the average of attributes by IoC for each of the organizations.

The used feeds provided on average 34 IoCs/day. The organization CIRCL was the main contributor for our dataset, providing 68.5% of the collected events. This was expected since CIRCL is the main entity behind the MISP project. Besides CIRCL, only CUDESO and CthulhuSPRL.be offered significant contributions, 10% and 19%, respectively. This means that these three organizations provided approximately 98% of the collected IoCs in our dataset.

B. Deduplicator Evaluation

To evaluate the capabilities of the deduplicator module, i.e., if it can eliminate duplicates according to the functionalities described in Section IV-A, we designed a set of tests that forced these situations. Table II shows the six different tests we defined. To execute the tests we extracted from our dataset a subset with 500 IoCs that contained identical, or near identical information as well as unrelated IoCs, and then we ran the deduplicator module. We verified that the module was able to identify and eliminate the duplicates (tests T1 to T4) while maintaining all other IoCs that are unrelated (tests T5 and T6), passing thus in all tests (column 3 of the table). This answers positively question 1.

TABLE II
TESTS FOR THE DEDUPLICATOR MODULE.

Test	Description	Passed
T1	Eliminates an IoC that is contained in another IoC	Y
T2	When analyzing two IoCs containing the same information, the IoC with the lowest trust level must be eliminated	Y
T3	When comparing two IoCs containing the same information and with the same trust level, the most recently created must be eliminated	Y
T4	When analyzing two IoCs containing the same information, the same trust level and the same creation date, the IoC with the highest ID must be eliminated	Y
T5	Do not eliminate an IoC that is highly similar, but not identical, to another IoC	Y
T6	Do not eliminate IoCs that are unrelated to other IoCs	Y

C. Enriched-IoC Generation

In this section we present the results obtained from running PURE with the n-level correlation method. We first present comparative results of running the method with different filters and afterwards we focus on our most restrictive filter.

1) Comparison of different filters: The first experiment we conducted was applying the n-level correlation method with different filters, obtaining thus different sets of interests of IoCs. For these experiments, the trust level was always set to 2, the option to include blacklists was disabled, while the type of connections varies. Table III presents these configurations (first 2 columns), the number of potential enriched IoCs obtained when applying different filters (column 3), and the minimum and maximum number of base IoCs that constitute them (last column).

The analysis of the *enr*-IoCs obtained with the less specific filters (all connections accepted and removal of the similarities based on network connections) showed that in most cases, the obtained *enr*-IoCs contained very tenuously connected IoCs. These filters generated 134 potential enr-IoCs, with some of them composed by up to 276 IoCs. We concluded that, when using such filters, the resulting IoCs do not have the most relevant information and they are very hard to analyze, i.e., there is not much benefit in aggregate IoCs with these criteria. On the other hand, the result of the third connection filter (last row of the table), i.e., the most restrictive filter, showed us that the final enr-IoCs are small (17 IoCs at maximum) and contain the most relevant and valuable information that characterize malicious threats in detail (presented next), which can be used by monitoring systems (e.g., SIEMs, IDS) to detect cyberattacks.

2) enr-IoCs with TI of quality: With the objective of verifying the quality of the *enr*-IoCs produced, we focus our analysis on the results of the third connection type. This experiment produced a total of 11 *enr*-IoCs obtained from the grouping of clusters of different sizes, containing from 2 to 17 distinct IoCs, each composed of between 5 and 4582 attributes and connect by 1 to 171 shared attributes. Table IV presents a description of the *enr*-IoCs obtained.

TABLE III
NUMBER OF POTENTIAL ENRICHED IOCS OBTAINED WHEN APPLYING DIFFERENT FILTERS.

Filter	Value	Enr-IoCs	IoCs
Trust level	2	- -	- -
Inclusion of blacklists	disable	- -	- -
Connection type	all connections	69	2; 276
	no network connections	65	2; 205
	only attacker, target or vulnerability	11	2; 17

The first feature we wished to ascertain when analyzing the *enr*-IoCs was if they showed an improvement in the completeness aspect of the quality of the threat intelligence produced. To consider that we were successful in providing this improvement we must verify if the IoCs grouped in the *enr*-IoCs all provide useful information on a specific topic, which is related to some specific attack or threat. This can be done by analyzing the topics each IoC deals in versus the topic covered by the majority of the IoCs of the *enr*-IoC, as well as by looking at the weight measurement indicator (*IoC-wgt*) of the connections established, as described in Section III-B. We must also guarantee that the enriched IoCs yield new additional information when compared with the base IoCs. This guarantee can be validated by analyzing the *Jaccard similarity* index.

When performing the analysis of the completeness of the new *enr*-IoCs *vis–a–vis* of their IoCs, we observed that indeed the association of IoCs was logical and, particularly for the *enr*-IoCs that aggregated more components, a dominant theme existed (as can be seen in Table IV). The theme was usually defined, as was to be expected, by the element that established the connection between the different components. This observation reinforces our remark that an increase in the specificity of the filters reduces the number of *enr*-IoCs but augments their quality. Also, the conjunction of selecting specific filters during the aggregation stage with the initial filtering performed during the set preparation is critical in the quality of the final product and in aiming the enrichment process.

Moreover, when analyzing our *enr*-IoCs, one observation that stood up was the potential they have in establishing connections between IoCs released at different times, allowing the creation of a timeline for the evolution of the exploration of vulnerabilities or of the behaviour of an attacker. As can be seen in columns 4 and 5, the most significant case where this occurs is the *enr*-IoC E11, which connects IoCs describing the usage of the vulnerability CVE-2012-0158 over a span of two and half years, from the end of October 2014 to the end of March 2017. However, we also see multiple other *enr*-IoCs that span over one year periods as is the case with the *enr*-IoCs E7, E8, E9 and E10. The creation of IoCs that provide a temporal overview is critical if we wish to create models to estimate the evolution in the actions of an attacker, or in understanding how the vulnerability is used over time after being first identified. Furthermore, we should refer that

TABLE IV
RESULTING ENRICHED IOCS BY USING N-LEVEL CORRELATION METHOD.

Enr-IoC	Num. IoCs	Characteristics	Earliest IoC date	Latest IoC date
E1	2	Composed of IoCs from the same organization connected by a WhoIs registrant email. It presents the connection between the Aveo malware campaign and an email used in the registration of multiple domains involved in other attacks	16-08-16	18-08-16
E2	2	Composed of IoCs from two distinct organizations that share 74 attributes and that both focus on reviews of information collected on an attacking group called 'Packrat'	08-12-15	09-12-15
E3	2	Composed of IoCs from the same organization that are related by a WhoIs registrant email and a domain belonging to that actor, connecting an event with domains used by that actor with the 'Multichair' campaign	08-09-15	15-04-16
E4	3	Composed of IoCs from two organizations that present different attacks where the same vulnerability was used CVE-2017-11882	04-12-17	25-01-18
E5	3	Composed of IoCs from two organizations connected by the use of the vulnerability CVE-2017-0262. This new IoC connects the activities of the APT 'Sofacy' with a campaign against financial institutions in Ukraine and an analysis of EPS processing zero-days attacks	09-05-17	21-02-18
E6	4	Composed of IoCs from two organizations it provides a connection between two previously unconnected APTs BlackVine and Cyber Kraken (aka Threat Group 3390/Emissary Panda), both active in 2015, who both used the ScanBox framework	27-10-14	05-08-15
E7	5	Composed of IoCs from one organization, it connects four different studies (of which only two are also connected when applying our filter) on the APT Sofacy (aka APT 28/Fancy Bear) via one central IoC dedicated to the use of a specific provider	21-04-16	21-09-17
E8	6	Composed of IoCs from two organizations, it is composed of a central cluster of three interconnected IoCs centered around the Turla (aka Snake/Uroburos) attacks and the APT Sofacy, to which the three other events are connected independently, bringing further information on the Turla attack, introducing an expansion of the Snake attack and, also, a framework called Cobra that was used by attackers also using Uroburos	13-11-14	18-08-16
E9	7	Composed of IoCs from two organizations, it is created around an IoC on the Neutrino Exploit Kit connecting via one of two vulnerabilities, CVE-2014-6332 and CVE-2013-2551, to multiple unconnected attacks and to another exploit kit	09-10-14	22-04-16
E10	11	Composed of IoCs from two organizations, it contains a group of ten interconnected events that focus on the activities of the APT Sofacy and allows the connection of an IoC on Operation Pawn Storm to them	23-10-14	26-09-16
E11	17	Composed of IoCs from three organizations, it contains a central core, composed of nine IoCs that are connected by the CVE-2012-0158, that branches out to three other groups of IoCs on campaigns and tools	30-10-14	29-03-17

TABLE V
enr-IOC E2 JACCARD AND *IoC-wgt* VALUES.

IoC X	#att IoC X	IoC Y	#att IoC Y	#att $X \cap Y$	Jaccard	*IoC-wgt*
2A	153	2B	133	74	0.346	0.518

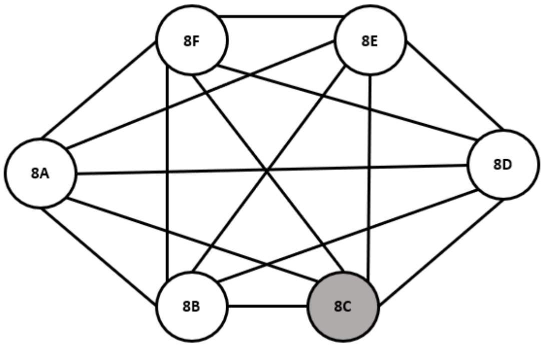

Fig. 2. Representation of *enr*-IoC E8.

this may potentially allow the establishment of connections between different attacks within the same campaign by an actor, that due to a change in targets or multiple attack times, are confused with distinct attacks, as reported by [16].

Next, to exemplify the increase in the completeness aspect of an IoC we present a detailed explanation of *enr*-IoCs E2 and E8.

Enr-IoC E2. In the case of E2, our analysis revealed the details of its composition and the values of its similarity indicators presented in Table V.

enr-IoC E2 is a clear example of the end product we want to obtain while allowing an easier analysis since it is only composed of two elements. In this case, the central theme of the *enr*-IoC is the threat actor Packrat and it aggregates two IoCs provided by CIRCL and CUDESO organizations. From Table V, we can observe that the *IoC-wgt* indicator for this *enr*-IoC is 52% from which we can infer that the number of attributes shared by the two- component IoCs is a sizable part of one or both components, so they are strongly connected. The values in the table also indicate that the *Jaccard* index is 35% allowing us to assert that the resulting *enr*-IoC gained a significant volume of information when compared to each of the two components when taken separately.

Enr-IoC E8. A more complex but also good example is provided by the *enr*-IoC E8 that focus on the Snake APT (aka Tesla or Uroburos). The components of this *enr*-IoC are all interconnected as can be seen in Fig. 2 which presents a graphical representation of the cluster of IoCs that form the *enr*-IoC (with the white circles representing IoCs originating from the feed Cthulhu.be and the grey circle the one originating from CIRCL). This interconnection is also visible in the indicators presented in Table VI. This *enr*-IoC presents an interesting characteristic: despite the fact that all its components share attributes, we can observe a group

TABLE VI
enr-IoC E8 Jaccard and *IoC-wgt* values.

IoC X	#att IoC X	IoC Y	#att IoC Y	#att $X \cap Y$	Jaccard	*IoC-wgt*
8A	52	8B	41	2	0.022	0.044
8A	52	8C	41	2	0.022	0.044
8A	52	8D	57	1	0.009	0.012
8A	52	8E	85	3	0.022	0.046
8A	52	8F	54	3	0.029	0.057
8B	41	8C	41	29	0.546	0.793
8B	41	8D	57	39	0.663	0.818
8B	41	8E	85	2	0.012	0.036
8B	41	8F	54	2	0.022	0.043
8C	41	8D	57	30	0.465	0.628
8C	41	8E	85	2	0.016	0.036
8C	41	8F	54	2	0.022	0.043
8D	57	8E	85	1	0.007	0.015
8D	57	8F	54	1	0.009	0.018
8E	85	8F	54	6	0.045	0.091

formed by the IoCs 8B, 8C and 8D that have a closer bond. This translates in higher values, when compared to the other relationships, of the *Jaccard* index and weight indicator for the connections among these three IoCs. In this sub-group, the IoCs share a higher percentage of their component attributes and as such the gain from their merging brings a lower increase in the number of new attributes added to each individual IoC. However, as the other relations show a low *Jaccard* index, there is still a significant increase in information that is added to the individual IoCs.

VII. Conclusion

In this paper, we proposed the PURE platform to improve the sharing of intelligence as provided by open source intelligence (OSINT) sources. PURE uses a novel cluster method, the *n-level correlation*, for clustering correlated indicators of compromise (IoCs) and generating high quality enriched IoCs. The proposed platform aims at improving the quality of the intelligence shared in its four vectors (timeliness, accuracy, relevance and completeness), by working on both its configuration and on the processing of the intelligence received by it. On the configuration side, care should be taken in the selection of adequate sources to provide a sound basis for the processing component of the solution. On the processing side, this solution proposes the addition of two novel modules, the *deduplicator* and the *enr-IoC generator*. The *deduplicator* module offers the capacity to eliminate duplicates from accumulating in the platform. The *enr-IoC generator* module performs three functions, the aggregation of associated IoCs in clusters, representing a single malicious threat, the correlation between IoCs within clusters, and the representation of these clusters as a single enriched IoC.

The platform uses the MISP platform to collect OSINT feeds and normalize this data, and to which was added the two modules. An experimental evaluation was performed with a dataset containing 1174 IoCs collected from 12 distinct organizations and 34 OSINT feeds. From this evaluation, we obtained 11 *enr*-IoCs with increased completeness when compared to an analysis of the individual IoCs.

Acknowledgements. This work was partially supported by the EC through funding of DiSIEM project, ref. project H2020-700692, and LASIGE Research Unit, ref. UID/CEC/00408/2019.

References

[1] Internet Society: Global Internet Report 2016 (Oct 16), www.internetsociety.org/globalinternetreport/2016/

[2] Cybersecurity Ventures: 2017 Cybercrime Report, https://cybersecurityventures.com/2015-wp/wp-content/uploads/2017/10/2017-Cybercrime-Report.pdf

[3] Chen P., Desmet L., Huygens C.: A Study on Advanced Persistent Threats. In: Proceedings of 15th IFIP TC 6/TC 11 International Conference. pp. 63–72 (Sep 2014)

[4] Symantec: Advanced Persistent Threat: A Symantec Perspective (2011), www.symantec.com/content/en/us/enterprise/white_papers/b-advanced_persistent_ threats_WP_21215957.en-us.pdf

[5] ENISA: Exploring the opportunities and limitations of current Threat Intelligence Platforms (2018)

[6] Kozuch I.: Cyber Threat Intelligence:How To Turn Quantity Into Quality (Apr 2018), https://www.peerlyst.com/posts/cyber-threat-intelligence-how-to-turn-quantity-into-quality-itay-kozuch

[7] Tounsi W. and Rais H.: A survey on technical threat intelligence in the age of sophisticated cyber attacks. In: Computers & Security. Vol 72 Number C pp. 212–233 (Jan 2018)

[8] Webroot: Threat Intelligence: What is it, and Can it Protect You from Today Advanced Cyber-Attacks? (2014), https://www.gartner.com/imagesrv/media-products/pdf/webroot/issue1_webroot.pdf

[9] SANS Institute: Threat Intelligence: What It Is, and How to Use It Effectively (2016), https://www.sans.org/reading-room/whitepapers/analyst/threat-intelligence-is-effectively-37282

[10] McKeon A.: Reduce Business Risk With an Effective Threat Intelligence Capability (Oct 2016), https://www.recordedfuture.com/threat-intelligence-capability/

[11] Caltagirone S.: The 4 Qualities of Good Threat Intelligence (Jul 2015), http://www.activeresponse.org/the-4-qualities-of-good-threat-intelligence/

[12] Dempsey M.E. : JP2-0, Joint Intelligence (2013), http://www.dtic.mil/doctrine/new_pubs/jp2_0.pdf

[13] Clemens Sauerwein C., Sillaber C., Mussmann A., and Breu R.: Threat Intelligence Sharing Platforms: An Exploratory Study of Software Vendors and Research Perspectives. In: Towards Thought Leadership in Digital Transformation: 13. Internationale Tagung Wirtschaftsinformatik pp. 12–15 (Feb 2017)

[14] Bank of England: CBEST Intelligence-Led Testing: Understanding Cyber Threat Intelligence Operations (2016), https://www.bankofengland.co.uk/-/media/boe/files/financial-stabilit/financial-sector-continuity/understanding-cyber-threat-intelligence-operations.pdf

[15] Meng G., Liu Y., Zhang J., Pokluda A. and Boutaba R.: Collaborative Security: A Survey and Taxonomy. In: ACM Comput. Surv. V 38 pages (Jan 2015)

[16] Liao X., Yuan K., Wang X.F., Li Z., Xing L., Beyah R.: Acing the IOC Game: Toward Automatic Discovery and Analysis of Open-Source Cyber Threat Intelligence. In: CCS16. pp. 63–72 (Oct 2016)

[17] Vacas I. and Medeiros I. and Neves N.: Detecting Network Threats using OSINT Knowledge-Based IDS. In: Proceeding of the 14th European Dependable Computing Conference. pp. 128–135 (Sept 2018)

[18] Alves F. and Bettini A. and Ferreira P. M. and Bessani A.: Processing Tweets for Cybersecurity Threat Awareness. arXiv:1904.02072 (Apr 2019).

[19] Loomis L.H. and Sternberg S.: Advanced Calculus - Revised Edition (1990), Jones And Bartlett Publishers

[20] MISP Project: MISP - Open Source Threat Intelligence Platform & Open Standards For Threat Information Sharing (2018), http://www.misp-project.org

[21] Jaccard P.: The Distribution of the Flora in the Alpine Zone. In: New Phytologist. pp. n. 2, vol 11, 37-50 (Fev 1912)