Generating Quality Threat Intelligence Leveraging OSINT and a Cyber Threat Unified Taxonomy

CLÁUDIO MARTINS, LASIGE, Faculdade de Ciências, Universidade de Lisboa - Portugal
IBÉRIA MEDEIROS, LASIGE, Faculdade de Ciências, Universidade de Lisboa - Portugal

Today's threats use multiple means of propagation, such as social engineering, email, and application vulnerabilities, and often operate in different phases, such as single device compromise, lateral network movement and data exfiltration. These complex threats rely on advanced persistent threats (APTs) supported by well-advanced tactics for appearing unknown to traditional security defences. As organisations realise that attacks are increasing in size and complexity, cyber threat intelligence (TI) is growing in popularity and use. This trend followed the evolution of APTs as they require a different level of response that is more specific to the organisation. TI can be obtained via many formats, being open-source intelligence (OSINT) one of the most common; and using threat intelligence platforms (TIPs) that aid organisations to consume, produce and share TI. TIPs have multiple advantages that enable organisations to quickly bootstrap the core processes of collecting, analysing and sharing threat-related information. However, current TIPs have some limitations that prevent their mass adoption. This paper proposes AECCP, a platform that addresses some of the TIPs limitations. AECCP improves quality TI by classifying it accordingly a *single unified taxonomy*, removing the information with low value, enriching it with valuable information from OSINT sources, and aggregating it for complementing information associated with the same threat. AECCP was validated and evaluated with three datasets of events and compared with two other platforms, showing that it can generate quality TI automatically and help security analysts analyse security incidents in less time.

CCS Concepts: • **Security and privacy → Intrusion detection systems**; **Domain-specific security and privacy architectures**; • **Information systems → Information extraction**; **Clustering and classification**.

ACM Reference Format:
Cláudio Martins and Ibéria Medeiros. 2022. Generating Quality Threat Intelligence Leveraging OSINT and a Cyber Threat Unified Taxonomy. 1, 1 (July 2022), 35 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In today's world, most organisations are digital, operating with technologies and processes of the Internet era. The changes in IT infrastructure and usage models, including mobility, cloud computing, and virtualisation, have dissolved traditional enterprise security perimeters, creating a vast attack surface for hackers and other threat actors [45]. Managing the digital landscape in which an organisation operates is a challenge that has never been more difficult, turning an organisation vulnerable to many forms of attack.

Not only the digital landscape has evolved, but there has also been a significant evolution in cyber threats, as adversaries have advanced their knowledge. They have deployed increasingly sophisticated means of circumventing individual controls within users' local environments and

Authors' addresses: Cláudio Martins, LASIGE, Faculdade de Ciências, Universidade de Lisboa - Portugal, claudio.dnm@gmail.com; Ibéria Medeiros, LASIGE, Faculdade de Ciências, Universidade de Lisboa - Portugal, imedeiros@di.fc.ul.pt.

XXXX-XXXX/2022/7-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

probed further into their systems to execute well-planned and orchestrated attacks [44]. With the increase of the digital landscape and the threat landscape complexity, organisations are more likely to be targeted and suffer a severe cyber-attack, with high financial and reputational impact. The high probability and impact of cyber-attacks, in addition to the significant regulatory pressure to protect the information, such as the European Union's General Data Protection Regulation, are encouraging organisations to look for new solutions to reduce their vulnerabilities [14].

One domain that has emerged during the past decade is cyber threat intelligence (CTI or TI for short). This new domain combines key aspects from incident response and traditional intelligence, and it can be defined as "the process and product resulting from the interpretation of raw data into information that meets a requirement as it relates to the adversaries that have the intent, opportunity and capability to do harm" [38]. However, compared to other cyber domains, such as incident response and security operations, TI is still in the early adoption phase, limited by the lack of suitable technologies, known as threat intelligence platforms (TIPs) [45][47]. Although organisations recognise the potentiality of TI, the lack of tools that would help them manage the collected information and convert it to actions is preventing the mass adoption of this kind of solution.

With the emergence of new threat actors, like the advanced persistent threats (APTs), organisations cannot rely on a single solution to protect from this type of threat. The static approach of traditional security based on heuristic and signature does not match new threats known to be evasive, resilient and complex. These complex threats rely on well-advanced tactics to appear unknown to signature-based tools and yet authentic enough to bypass spam filters [16]. Today's organisations must deploy a multi-layered defence to improve their chances of detecting or disrupting an attack to fight these threats.

Under a form of open source intelligence (OSINT), TI information can provide knowledge to a vast selection of systems and processes that form this multi-layered defence, such as anti-virus and intrusion prevention systems and the processes that manage these solutions and review the events generated by them. This knowledge can be collected from many sources using threat intelligence platforms (TIPs). However, TIPs receive thousands of security events, which makes it hard to analyse them to extract relevant data about threats. The volume and quality of data are the most common barriers to effective information exchange. Also, shared data is often outdated and not specific enough to aid the decision-making process, becoming unactionable [48]. The confidence level of information is another barrier since most sources do not provide this information, forcing security analysts (SOC) to put additional effort into evaluating and verifying the received data. Also, most organisations cannot make valuable use of their threat data because there is too much, approximately 250 to millions of indicators of compromise (IoC) per day [48]. Considering the volume of shared threat information, most of the platforms end up being data warehouses rather than platforms where threat information can be analysed. Moreover, the time-consuming SOC analysts spent analysing and classifying incidents have increased due to this volume of data, not valuable information and duplication of incident classification in several public incident taxonomies (e.g., eCSIRT and ENISA). There are few platforms [3][18][1] that deal with these drawbacks. They aggregate diverse OSINT data related to the same threat into a single event. At first, this approach is beneficial, as it avoids the manual analysis of several individual events and the manual attempt to establish their relationships. As a result, it will decrease the time spent by SOC analysts performing this task. However, on the other hand, aggregating a set of events into one will increase the amount of information that analysts must check. This amount can reach more than a thousand attributes in an event, and, therefore, the time required to analyse it can be longer than the time needed to analyse the set of events individually.

This paper proposes the *Automated Event Classification and Correlation Platform* (AECCP) that implements an approach to address some of TIPs limitations by generating highly information-rich objects under a standard format and a *single unified taxonomy* (*unified taxonomy* (UT), for short), with their threat categories characterised by *main threat attributes*. Also, it correlates and aggregates these objects into clusters of objects, generating thus quality TI that shares the same threat type and other information. To improve the collection and automatic classification of actionable TI, as well as to define the UT, we first need to understand the TI life cycle, the available information sources and current TIPs, and to identify the *main attributes* that allow characterising each threat category of UT. This requires working on all levels of the intelligence-gathering operation, using an automated system to (i) receive data from multiple sources, (ii) improve the enrichment process and validate the information collected by cross-referencing it, (iii) produce objects under a standard format and taxonomy, (iv) store the obtained intelligence in such a way that it can be applied in the optimisation of defence mechanisms. Moreover, by using a UT and the main threat attributes, the problem that arose from the platforms aforementioned will be solved.

This paper is the first to (i) propose a unified taxonomy to classify security events, (ii) study and identify the main attributes that better describe threat types, (iii) classify security events automatically into an incident category and removes the overlap of classification tags, without human intervention, and (iv) propose a platform to reduce the amount of information aggregated in a single event, after an event correlation and clustering task. Moreover, our approach aims to improve the response of threat analysts and all the systems used by the organisation against today's complex threats. In addition, it aims at finding ways to benefit from OSINT to increase the detection capabilities of defence mechanisms, such as security information and event management systems (SIEMS) or intrusion detection systems (IDS), reducing the number of false positives and negatives.

We validated and evaluated AECCP with three datasets of security events. Our results suggest that AECCP can automatically classify TI into an incident category and generate new and enriched TI that associate different security events regarding the same threat in a single way. Also, we compared AECCP with two platforms from literature, and the results show that our approach performs better than the others.

The main contributions of the paper are: (1) a single unified taxonomy to reduce the overlapping of taxonomies with the same meaning and simplify the event classification while maintaining its details. (2) the identification of the main attributes that characterise each incident category into the proposed taxonomy, which will allow reducing the volume of shared information. (3) an approach that aims to improve quality threat intelligence produced by TIPs by automatically classifying and enriching it. The approach is composed of a set of modules, each one focused on one or more limitations of TIPs and verified in our data analysis. (4) the AECCP and its assessment with three event datasets and two other platforms.

2 BACKGROUND AND RELATED WORK

2.1 Advanced Persistent Threats

Today's generation threats are multi-vectored and often multistage, i.e., most attacks use multiple means of propagation, such as social engineering, email, and application vulnerabilities, and most attacks operate in different phases, such as single device compromise, lateral network movement, and data exfiltration [48]. These complex threats rely on social engineering techniques, the latest zero-day vulnerabilities, and well-advanced tactics for appearing unknown to signature-based tools and yet authentic enough to bypass spam filters. Traditional security defences were developed to inspect each attack vector as a separate path and each stage of an attack as an independent event, failing in identifying and analysing an attack as an orchestrated series of cyber incidents [16].

The advanced persistent threats (APT), being one of today's generation threats that had a significant impact on the rise of cybercrime, branched from young hackers in the Black Hat community, whose objective was mayhem and reputation, to organised crime groups provided by states and private entities [45]. Chen et al. characterise APTs and separate them from other criminal enterprises online, being them: specific targets and clear objectives, highly organised and well-resourced attackers, long-term campaigns with repeated attempts, and stealthy and evasive techniques [5].

2.2 Open Source Intelligence (OSINT)

The earliest forms of open-source intelligence (OSINT) dates back to the Second World War, marked by the ability to find relevant information and combine it in a way that treats information as a resource rather than a commodity [23][17]. OSINT can be defined as intelligence produced from publicly available information (open-source information, OSINF), such as information gathered from radio, television, newspapers, websites, blogs, papers, conferences, etc. Nowadays, due to the development of the Internet, this type of information has become significantly more accessible and cheaper to gather than the traditional public information acquired by clandestine services. In comparison to other sources of information, like human intelligence, OSINF can sometimes provide extra information and be a more reliable and safe way of acquiring intelligence [11].

To produce OSINT, OSINF is analysed, edited, filtered and validated. Moreover, the information gathered is linked with other sources to verify, complement, and contextualise the collected data. The more public available sources, the better intelligence will be produced [17][11]. OSINT has become one of the most common forms of intelligence and is considered a goldmine for organisations [36]. For instance, recent studies stated that valuable and early information can be provided by social networks, such as Twitter [48][39]. One of the biggest advantages of using OSINT is the cost, as it is much less expensive than traditional information-gathering tools. Additionally to the cost advantage, OSINT has many benefits when it comes to sharing and accessing information, as this latter can be legally and easily shared with anyone, and open sources are always available and up to date [19]. However, OSINT has some constraints, such as the high quantity of available information that needs to be processed to create valid intelligence, demanding a high amount of work to extract useful information from the noise. This task requires a large amount of analytical work from security specialists to distinguish valid, verified information from false, misleading or inaccurate data. A final constraint of OSINT is that its production may not always provide the needed answer since it only uses available information [19].

2.3 Threat Intelligence (TI)

Threat intelligence (TI) can be defined as "evidence-based knowledge, including context, mechanisms, indicators (...) about the hazard to assets that can be used to inform decisions regarding the subject's response to that menace or hazard" [50].

In its simplest form, TI is the process of understanding the threats towards an organization based on available information. However, there must also be an understanding of how the information relates to the organization. Hence, it must be combined with contextual information to determine relevant threats to the organization. Moreover, TI is valuable to an organization only if it is actionable. If the Security Operation Center (SOC) cannot determine how to best respond, combat or mitigate a threat to the organization, then the information provides little to no value [4]. Detecting incidents sooner and potentially even preventing them is the overall goal of TI. Organizations often see TI as a way to reinforce the environment and prepare for both known and unknown threats.

TI has grown in popularity and use amongst organizations as they realize that attacks have increased in size and complexity. According to a CTI survey, 85.5% of respondents have at least one

person responsible for consuming or producing TI in their organization and 7.1% of respondents plan to have one shortly. This trend followed the evolution of targeted attacks and APTs as they require a different level of response that is more specific to the organization [21]. Many organizations are convinced that TI is a valuable tool to help them better understand their attackers.

As we stated, the objective of creating TI is the creation and delivery of a product that can be acted upon. While threat intelligence professionals find value in sharing threat information through informal and traditional communication channels, the results are inconsistent and unscalable. Hence, better frameworks were needed for communicating TI to provide an adequate answer to today's complex threats. Such frameworks should include: standardized reporting terminology and processes; benefit in information sharing for cybersecurity purposes; the ability for users to create trusted communities; and, technical infrastructure to share and analyze TI at machine speed. In the absence of an industry-standard framework, current sharing mechanisms include: private or restricted face-to-face meetings and phone calls; emails, forums and message boards; web portals with wiki-type capabilities; web portals acting as document management systems; web portals (some with APIs) allowing downloads of structured data; and, web portals offering social networking facilities with secure access and sharing controls [12].

TI represents security threat activities that are provided as a form of indicators of compromise (IoC), i.e., information artefacts obtained from a forensic analysis that aggregate data on malicious activity in a system or within a network that was attacked [26]. For sharing TI among entities and security platforms and structuring its information, diverse standard formats have been proposed, being OpenIoC [9], STIX [32], TAXII [33], CSV, and MISP format the most popular. However, its use is not widespread and poorly implemented [37].

2.4 Threat Intelligence Platforms (TIP)

Threat intelligence sharing platforms (in short, threat intelligence platforms or TIPs) was introduced to fill the industry-standard gap in TI sharing, and gaps and limitations of actual detection and monitoring defence mechanisms placed in IT infrastructures [46]. In this sense, TIPs are used for OSINT and TI collection and their processing, storage, sharing, and integration of their resulting data with other security platforms and tools related to incident response and threat management (e.g., SOC, CSIRTs). They retrieve (structured and unstructured) data from several external sources (e.g., OSINT feeds) and process these data by applying various operations, such as filtering, normalization, aggregation, and some correlation [3].

TIPs usually vary in the (1) objective: some are used to operational information while others may be focused on long-term risk analysis, (2) the scope of their action: from accepting only processed inputs to possessing natural language processing capacities, and (3) their capabilities: current platforms range from data acquisition and storage to advanced analytics using machine learning. Despite their differences, the functionalities of TIPs follow the steps of the threat intelligence life cycle, namely planning and direction, collection, processing and exploitation, analysis and production, dissemination and integration [4][20][25][34].

Since TIPs existence, their adoption by organizations has grown and played an important role in spreading security threat activity among the collaborative entities working in this field. However, their adoption and implementation are still in their infancy [43], having many limitations to be resolved, e.g., automatic trust assessment and classification of TI and advanced capabilities of analysis, where SOC intervention continue to be required to filter and retrieve TI information that is relevant and effectively actionable.

There are some open-source TIPs that have been adopted by organizations, being the next four those widely used [48]: MISP (the Malware Information Sharing Platform) [30], CIF (the Collective

Intelligence Framework) [8], CRITs (the Collaborative Research Into Threats) [31], and SoltraEdge [22]; and being MISP the most popular.

2.5 MISP

MISP was initially created by the NATO Computer Incident Response Capability Technical Centre (NCIRC TC) to implement the Smart Defence concept and, presently, is owned by the Computer Incident Response Centre Luxembourg (CIRCL). One of the key concepts of MISP is the sharing of intelligence among members of the same community [49][30].

Currently, MISP has not only, but mainly, the following capabilities: sharing; storage; automatic correlation of indicators of compromise (IoCs); advanced filtering capabilities; export and import of data in the most popular formats, namely STIX, OpenIOC, CSV and MISP standardized format [49][10]. IoCs, also called MISP events, contain technical and general information of TI, which are represented in MISP format and stored in a database of indicators.

A new entry in MISP's database is called an event object, which can be defined as a set of characteristics and all kinds of descriptions of an IoC. These characteristics and relevant information are called attributes. Examples of attribute types are hash, filename, hostname and IP address. An attribute can even be a complex object that contains multiple attributes. An example of a complex attribute is an anti-virus signature, which can include the name of the anti-virus, the name of the signature, and the detection date [49]. Furthermore, each attribute can be correlated with other simple or complex attributes. Also, IoCs, when stored, are automatically correlated to describe the relationships between attributes and indicators [10].

2.5.1 Taxonomies. Data classification is often bound to internal, community or national classification schemes. One common problem is the mapping of events into categories. It is a complex task since categories are not always known in advance. Since a centralised pre-defined set of definitions that satisfies all the potential users is a hard challenge, MISP uses a distributed approach based on machine tags. However, the freedom of defining tags can easily lead to a situation where multiple tags have the same meaning, making filtering complicated. A new concept of tagging was introduced to overcome this problem – the taxonomies. Taxonomy is based on a triple tag structure with a namespace, a predicate and a value, for example, `[enisa:nefarious-activity-abuse="ransomware"]`. This flexible concept allows classifying and tagging events following an organisation own classification schemes or existing taxonomies used by other organisations. A clear advantage of this concept is the still human-readable format of the machine tags [49].

In its default configuration, MISP includes a set of public incident classification taxonomies [29], where some of the most used of them are described next, and their tags are presented in Table 1 as being recognised in the MISP tag structure.

- *eCSIRT.net* [7] (middle-high of column 1). This taxonomy was developed many years ago, but the main categories are still current and can easily be used. On the other hand, the subcategories can lead to problems with classifying an incident. Despite its defects, many European Computer Security Incident Response Teams (CSIRTs) use it, which allow teams to team up with others.
- *CIRCL.LU* [6] (middle-bottom of column 1). MISP owners and main contributors use their taxonomy for classifying incidents. With some similarities with eCSIRT.net taxonomy, CIRCL.LU only has one level of classification.
- *Microsoft implementation of CARO Naming Scheme* [27] (second column). According to the Computer Antivirus Research Organization (CARO) malware naming scheme, Microsoft designates malware and unwanted software. This scheme was created by a committee at CARO and was the first attempt to make malware naming consistent.

Table 1. eCSIRT.net, CIRCL.LU and Microsoft implementation of CARO taxonomies recognized in the MISP tag structure.

eCSIRT.net taxonomy main category	**Microsoft implementation of CARO Naming Scheme**
ecsirt:abusive-content	ms-caro-malware:malware-type="Adware"
ecsirt:malicious-code	ms-caro-malware:malware-type="Backdoor"
ecsirt:information-gathering	ms-caro-malware:malware-type="Behavior"
ecsirt:intrusion-attempts	ms-caro-malware:malware-type="BroswerModifier"
ecsirt:intrusions	ms-caro-malware:malware-type="Constructor"
ecsirt:availability	ms-caro-malware:malware-type="DDoS"
ecsirt:information-content-security	ms-caro-malware:malware-type="Dialer"
ecsirt:fraud	ms-caro-malware:malware-type="DoS"
ecsirt:vulnerable	ms-caro-malware:malware-type="Exploit"
ecsirt:other	ms-caro-malware:malware-type="HackTool"
ecsirt:test	ms-caro-malware:malware-type="Joke"
	ms-caro-malware:malware-type="Misleading"
CIRCL.LU taxonomy	ms-caro-malware:malware-type="MonitoringTool"
circl:incident-classification="spam"	ms-caro-malware:malware-type="Program"
circl:incident-classification="system-compromise"	ms-caro-malware:malware-type="PUA"
circl:incident-classification="scan"	ms-caro-malware:malware-type="PWS"
circl:incident-classification="denial-of-service"	ms-caro-malware:malware-type="Ransom"
circl:incident-classification="copyright-issue"	ms-caro-malware:malware-type="RemoteAccess"
circl:incident-classification="phishing"	ms-caro-malware:malware-type="Rogue"
circl:incident-classification="malware"	ms-caro-malware:malware-type="SettingsModifier"
circl:incident-classification="XSS"	ms-caro-malware:malware-type="SoftwareBundler"
circl:incident-classification="vulnerability"	ms-caro-malware:malware-type="Spammer"
circl:incident-classification="fastflux"	ms-caro-malware:malware-type="Spoofer"
circl:incident-classification="sql-injection"	ms-caro-malware:malware-type="Spyware"
circl:incident-classification="information-leak"	ms-caro-malware:malware-type="Tool"
circl:incident-classification="scam"	ms-caro-malware:malware-type="Trojan"
circl:incident-classification="cryptojacking"	ms-caro-malware:malware-type="TrojanClicker"
circl:incident-classification="locker"	ms-caro-malware:malware-type="TrojanDownloader"
circl:incident-classification="screenlocker"	ms-caro-malware:malware-type="TrojanDropper"
circl:incident-classification="wiper"	ms-caro-malware:malware-type="TrojanNotifier"
circl:incident-classification="sextortion"	ms-caro-malware:malware-type="TrojanProxy"
	ms-caro-malware:malware-type="TrojanSpy"
	ms-caro-malware:malware-type="VirTool"
	ms-caro-malware:malware-type="Virus"
	ms-caro-malware:malware-type="Worm"

2.6 Limitations of Threat Intelligence Platforms

TIPs have multiple advantages that enable organisations to easily bootstrap the core processes of collecting, normalising, enriching, correlating, analysing, disseminating and sharing threat information. However, current solutions have some limitations that prevent their mass adoption. Next, we present the limitations related to the current state and usage of TIPs [13] [47][35].

- *LT1 - Shared threat information is too voluminous.* One of the problems is the overload of threat information shared via open-source, commercial sources and communities. Combining shared threat information from different sources makes the relevant intelligence hard to find and makes it difficult to generate value.
- *LT2 - Limited technology enablement in threat triage.* There is limited technology enablement to facilitate the relevancy determination process. Currently, this process is done manually, in a complex way, and dependent on the analyst.
- *LT3 - Data Quality.* The confidence level of information is not provided by most of the feed, forcing analysts to put additional effort into evaluating and verifying the received data.

- *LT4 - Limited analysis capabilities.* Most TIPs have limited capabilities related to browsing, attribute-based filtering, advanced searching information, pivoting, exploration and visualisation.
- *LT5 - Limited advanced analytics capabilities and automation tasks.* Most TIPs have limited capabilities related to aggregation, composition, generalisation of data, as well as the capability to de-duplicate, tag and classify data automatically.
- *LT6 - Focus on data collection.* Considering the volume of shared threat information and the limited analysis capabilities provided by TIPs, most of the platforms end up being data warehouses rather than platforms where threat information can be shared and analysed.
- *LT7 ? Limited threat knowledge management.* No common vocabulary is used for describing threat actors, tactics, techniques, procedures and tools.
- *LT8 - Focus on tactical IoCs.* Tactical indicators of compromise are mostly shared, lacking comprehensive threat information. Standardised formats are underused or even not used during information sharing, noting that most information is exchanged in unstructured files.
- *LT9 - Trust related issues.* Most TIPs have limitations in the way that organisations interact and contribute to specific communities, and most platforms do not allow organisations to share only specific types of threat data with particular communities.
- *LT10 - Diverse data formats.* While there are community efforts to provide connectors between different standards and formats, converting information without losing any elements or context from the source format is a challenge. Most TIPs tend to stay with one format, limiting the flexibility of the TIP users.
- *LT11 - Shared intelligence without expiration date.* Currently, the time-to-live information is not provided by most of the feeds, and TIPs have limited capabilities in handling this type of metadata information.
- *LT12 - Diverse APIs and requirements for integration.* TIPs integrate with a standard set of services and tools while the owners prioritise requests for additional integrations.
- *LT13 - Limited workflow enablement.* Currently, TIPs provide limited workflow capabilities that would make the process of threat management more efficient, such as the capability of stakeholders to send requests for information.

2.7 Platforms for Resolving Limitations of TIPs

A few platforms try to reduce some TIPs' limitations and improve the TI processing.

PURE [3] is a platform that generates improved intelligence based on OSINT. This enhanced intelligence translates into new enriched IoCs obtained by correlating and combining IoCs from different OSINT feeds sharing information about the same threat. The novel cluster method used by PURE allows the creation of clusters that can be summarised and converted into an enriched IoC, allowing the discovery of unidentified patterns and the detection of new complex attacks. The platform comprises the normalisation of the different IoC formats in a single one and compares the IoCs received with the IoCs stored in the database to check the existence of duplicates. Besides discarding the duplicated IoCs, it also discards those that provide no new information. The set of IoCs of interest resulting from a filter step is sent to a clustering module, which applies similarity and weighs metrics over them to aggregate similar and related IoCs to create quality TI. IoCs belonging to a cluster are correlated to find the most relevant information that characterises a threat and then converted into a single enriched IoC.

ETIP [15] [18] is a platform that extends the importing capabilities, the quality assessment processes and the information-sharing capabilities in current TIPs. ETIP gathers and processes structured information from external sources, such as OSINT and a monitored IT infrastructure. It comprises two main modules: a composed IoC module, in charge of collecting, normalising,

processing, and aggregating IoCs from OSINT feeds; and a context-aware intelligence sharing module, able to correlate, assess and share static and real-time information with data obtained from multiple OSINT sources. ETIP computes a threat score associated with each IoC before sharing it with other tools and trusted external parties. Enriched IoCs produced by ETIP contain a threat score that allows SOC analysts to prioritise the analysis of incidents. The threat score evaluates heuristics with two weights: individual weights assigned to every attribute based on their relevance, accuracy and variety, and; a global weight (i.e., completeness criterion) assigned to the heuristic. The higher the threat score value, the more reliable the IoC.

SYNAPSE [1], a Twitter-based streaming threat monitor for threat detection in SOCs, implements a pipeline that gathers tweets from a set of accounts, filters them based on the monitored infrastructure, and classifies the remaining tweets as either relevant or not. The pipeline is composed of a data collector, a filter, pre-processing and feature extraction module, a classifier, and a clustering module. The data collector requires a set of accounts, from which it will collect every posted tweet using Twitter's stream API. The filtering approach assumes that a tweet must mention a particular IT infrastructure asset when referring to a threat to a specific IT infrastructure asset. Only tweets that include at least one of the keywords will pass the filter. The pre-processing and feature extraction module is then used to normalise the tweet representation before the classifier. Two classifiers were explored for the classification of tweets according to their security relevance: Support Vector Machines (SVM) and Multi-Layer Perceptron (MLP) Neural Networks. Finally, SYNAPSE uses clustering to aggregate similar tweets in the news feed stream, adapting a Clustream algorithm to achieve the desired threat aggregation. Relevant tweets are grouped in dynamic clusters and presented as IoCs that can be manually inspected or fed to SIEMs and other TI tools.

Table 2 presents which TIPs limitations (stated in Section 2.6) are addressed by these platforms (columns 3 to 5). They all have the main objective of creating quality TI through new analytical approaches and in an automated way. The new TI is obtained by filtering and combining OSINT associated with the same threat in a single security event. The concretisation of this objective addresses the first six TIPs' limitations (LT1 to LT6) since the resulting TI will allow decreasing the amount of individual and not related data (security events) that SOC analysts must analyse. However, on the other hand, this resulting TI as aggregates in a single event much more information (the merging of several events) than those contained in individual events, the task to analyse this quantity of data by SOC analysts can be more challenging. PURE and ETIP also deal with LT10 because they can receive OSINT in diverse formats. As ETIP consumes data from the organisation's IT infrastructure to analyse it jointly with OSINT and the resulting TI can be exported to be used in defence mechanisms, it deals with LT8 and LT11, respectively. In turn, SYNAPSE also addresses LT11 for the same reasons as ETIP.

Table 2. The TIPs limitations addressed by PURE, ETIP, SYNAPSE and AECCP platforms.

ID	Limitation	PURE	ETIP	SYNAPSE	AECCP
LT1	Shared threat information is too voluminous	x	x	x	x
LT2	Limited technology enablement in threat triage	x	x	x	x
LT3	Data Quality	x	x	x	x
LT4	Limited analysis capabilities	x	x	x	x
LT5	Limited advanced analytics capabilities and tasks automation	x	x	x	x
LT6	Focus on data collection	x	x	x	x
LT7	Threat knowledge management limitations				x
LT8	Focus on tactical IoCs		x		x
LT9	Trust related issues				x
LT10	Diverse data formats	x	x		x
LT11	Shared intelligence without expiration date		x	x	x
LT12	Diverse APIs and requirements for integration				
LT13	Limited workflow enablement				

The platform we propose – AECCP (last column of the table) – addresses all TIPs' limitations, except the last two (LT12 and LT13). Although AECCP shares the main objective of the other platforms, it employs different types of analysis for filtering and combining data (detailed in Section 4). It gives a step further by proposing a UT and threat main attributes to classify OSINT data, which both will allow reducing the amount of information consolidated in a single and resulting event (something that the other platforms face), and, therefore, decrease the effort that SOC analysts must employ in analysing such data. These valencies will treat the limitations of LT7 and LT9 and make AECCP the first platform that achieves that. Also, it is the first platform that classifies security events in incident categories and removes the existent overlap of classification of public taxonomies' tags without human intervention, i.e., automatically. In addition, our platform consumes diverse OSINT data formats (LT10) and external data (LT8) to improve the quality of TI, and the generated TI can be shared and used in organisations' defence mechanisms (LT11).

3 DATA ANALYSIS FOR AN UNIFIED TAXONOMY AND THREAT MAIN ATTRIBUTES

As we stated before, the primordial goal of this work is to address some of the limitations of TIPs, described in Section 2.6. We manage all of them, except the last two (L12 and L13), focusing on the first seven limitations. More specifically, we aim to solve those related to the processing of data in the platforms, i.e., classify, analyze, and generate data automatically, minimizing thus the human intervention in this process. However, to produce the most accurate and complete TI, we have to consider resolving the other four limitations since they are related to these seven. For example, to obtain more comprehensive data about a given attack, it is needed to consider and process OSINT data that can come in diverse formats (L10). To address the limitations with an adequate solution capable of treating and minimizing them, first, we had to understand such constraints. Hence, this section presents the data analysis performed to obtain such understanding.

The analysis is based on MISP events, as MISP is the most open-source TIP adopted among organizations. Therefore, the section firstly gives an overview of the data sources used to collect the events and how the dataset used in the analysis was built (presented next). Secondly, it presents an analysis of MISP taxonomies, which shows how the vast set of public incident classification schemes included in MISP to classify the same threat can increase unnecessary complexity and relevant information. To tackle this and decrease such unnecessary information, we propose a *single unified taxonomy* which is defined in Section 3.2. In addition, an analysis of MISP event attributes is provided, showing that too many attributes in a single event can also increase the unnecessary complexity, specifically if they do not add useful information. To face this problem, we propose a solution in Section 3.3 that involves discovering which are the *most prevalent attributes that underlie a threat*. Finally, a brief explanation on how we can take advantage of references to external platforms to increase the quality of TI is given in Section 3.4.

3.1 Data Sources and Dataset

The source information to get the dataset for analysis was provided from external OSINT feeds and the TIP to collect and process them was MISP. MISP can process different feed formats, namely MISP standardised format, CSV and free text. CSV and free text feeds are only parsed as MISP Attributes and do not take advantage of all MISP functionalities. Contrarily, the MISP formatted feeds can be parsed from simple MISP Attributes to the more complex MISP Objects and benefit from all MISP functionalities. Therefore, we left aside CSV and free text feeds and worked only

with MISP formatted feeds, resulting thus in the following three feeds: CIRCL OSINT Feed[1], The Botvrij.eu Data[2], and inThreat OSINT Feed[3].

From these three feeds, we collected 1,366 events published by 14 different organisations, such as CIRCL, CUDESO, InThreat, CthuluSPRL.be, Synovus Financial, VK-Intel, ESET and NCSC-NL. However, some of these events are dated to 2014, near the embryonic phase of MISP, meaning poorer events with minimal information and more events containing collections of IoCs from multiple attacks (e.g., blacklists). In contrast, recent events (since 2016) were richer in knowledge, and many more events corresponded to one attack. Consequently, we shortened the initial dataset only to contain richer events, resulting in 1,168 out of 1,366 events, in which most of them were provided by CIRCL and CUDESO with 907 and 120 events, respectively.

3.2 Unified Taxonomy

Over the past decades, multiple cyber threat classification systems have been proposed, some of them focus on the classification of actors and methods [35], while others focus on specific techniques [28] or specific targets [40]. With more than 100 classification systems, this complex array of taxonomies adds confusion when a security analyst manually analyses a threat and, consequently, increases the time and effort he spends. This complexity is increased in MISP with unnecessary information since an event can be classified by the analyst for a given incident with different taxonomies, meaning that that event will have several tags with the same mean. For example, an event classified as *ransomware* has five tags mapping different taxonomies, namely *[ecsirt:malicious-code="ransomware"]*, *[malware_classification:malware-category="Ransomware"]*, *[veris:action:malware:variety="Ransomware"]*, *[enisa:nefarious-activity-abuse="ransomware"]*, and *[ms-caro-malware:malware-type="Ransom"]*. Based on this evidence, in this section, we present a solution to reduce this complexity by proposing a *single unified taxonomy* (UT).

As previously explained, events in MISP are classified with tags following taxonomies, meaning that a classified event requires having at least one tag. Our dataset based on this principle contains 1166 tagged events and 2 untagged events. However, a more detailed analysis showed that many of the tagged events did not have a tag that allowed to classify them correctly into an incident category. Only 691 (out of 1166) events were tagged into an incident category. Furthermore, we found that several occurencies had multiple overlapping classification tags from different taxonomies, meaning duplicated information about their type.

From the 1166 tagged events, 493 different tags were extracted. Table 3 shows the 16 most used tags in their classification. A more extensive table can be found in Appendix A [24]. From the extracted tags, only 13% of them (62) corresponded to a known incident classification taxonomy (ID 4-6), meaning that most remaining tags did not add information about the type of the threat but added information about its source (IDs 2, 8, 9 and 14) and its sharing, such as the Traffic Light Protocol (TLP) and OSINT (IDs 1 and 3). Additionally, 61% of the tags (i.e., 302) corresponded to MISP Galaxies. MISP Galaxies are highly customizable and can correspond not only to known attacks (ID 7) but also to attack patterns, threat actors (ID 11) and tools (ID 13). Therefore, we opted not to consider MISP Galaxy tags and the other tags referred to above as classification tags due to the high heterogeneity and low information about the type of threat they carried. Hence, for further analysis, we only considered the 62 tags associated with incident classification, which belong to 10 different incident classification taxonomies (first 10 IDs of Table 4).

The UT we propose is based on structures of eCSIRT.net incident taxonomy and CARO malware naming scheme, and it aims to simplify the event classification while maintaining its details. Also,

[1]https://www.circl.lu/doc/misp/feed-osint/
[2]http://www.botvrij.eu/data/feed-osint/
[3]https://feeds.inthreat.com/osint/misp/

Table 3. The 16 most used tags in events.

ID	Tag	Hits	ID	Tag	Hits
1	tlp:white	1133	9	osint:source-type="block-or-filter-list"	32
2	osint:source-type="blog-post"	275	10	circl:topic="finance"	31
3	Type:OSINT	273	11	misp-galaxy:threat-actor="Sofacy"	26
4	circl:incident-classification="malware"	218	12	OSINT	26
5	malware_classification:malware-category="Ransomware"	113	13	misp-galaxy:tool="Trick Bot"	24
6	ecsirt:malicious-code="ransomware"	98	14	osint:source-type="technical-report"	23
7	misp-galaxy:ransomware="Locky"	70	15	workflow:todo="expansion"	22
8	inthreat:event-src="feed-osint"	32	16	osint:lifetime=ephemeral	21

Table 4. The 10 taxonomies used for incident classification and the 22 of taxonomies analyzed to define the unified taxonomy.

ID	Taxonomy	ID	Taxonomy
1	CIRCL.LU taxonomy	12	Information security indicators from ETSI GS ISI
2	eCSIRT.net incident taxonomy	13	Malware Attribute Enumeration and Characterization (MAEC)
3	ENISA threat taxonomy	14	Reference Security Incident Classification Taxonomy
4	Microsoft implementation of CARO Naming Scheme	15	Threats targetting cryptocurrency, based on CipherTrace report.
5	Internal taxonomy for Canadian Centre for Cyber Security (CCCS)	16	Open Threat Taxonomy
6	Europol common taxonomy for law enforcement and csirts	17	Penetration test (pentest) classification
7	Vocabulary for Event Recording and Incident Sharing (VERIS)	18	Infoleak taxonomy
8	ENISA threat taxonomy in the scope of securing smart airports	19	Common Taxonomy for Law enforcement and CSIRTs
9	SANS malware classification based on "Malware 101 – Viruses"	20	MONARC Threats Taxonomy
10	CERT-XLM Security Incident Classification	21	Distributed Denial of Service - or short: DDoS - taxonomy
11	GSMA - Fraud and Security Group	22	Incident disposition based on NASA Incident Response and Management Handbook

since most taxonomies have two tiers of classification, such as the eCSIRT.net incident taxonomy, we opted to follow this level of detail. This allows us to choose the granularity level of the classification. To define UT we analyzed the 22 public taxonomies listed in Table 4, for the tags related to incident classification[4]. UT is composed of 8 incident categories of Tier 1 (such as the other two taxonomies) and 38 sub-categories of Tier 2 distributed by Tier 1 categories.

Table 5 resumes how each public taxonomy of Table 4 contributed to the definition of UT, in terms of number of incident classification tags for each Tier 2 sub-category (column 3), and so, how many taxonomies are in root of each Tier 1 and Tier 2 (column 26). In total, 354 tags from public taxonomies were mapped to our taxonomy, being *Veris*, *CARO* and *Europol* the taxonomies that most contributed (line 41). Also, *eCSIRT.net*, *Veris*, *CERT-XLM*, and *CARO* were the taxonomies that most participated in the definition of Tier 2 sub-categories (last line).

Table 6 contains an excerpt of UT, showing the relationship map we created for all public taxonomies (columns 1 to 3). The complete definition of UT can be found in Appendix B [24].

Additionally, a bag of words was defined for each Tier 2 of UT to describe them and allow further classification. Each bag was created based on words extracted from the public taxonomies and synonyms from these words. These bags of words will not only support further analyses over events with public taxonomy tags but, most importantly, be used to analyse events without public taxonomy tags, e.g., those two untagged events from our dataset that were not classified yet. The last column of Table 5 contains the number of words affected to each category, in a total of 147 words, and the last column of Table 6 presents the bag of words mapped by category of UT. The complete list of bags of words can be found in Appendix B as part of the definition of the UT [24].

3.3 Main Threat Attributes

As previously stated, the volume of shared information is one of the TIPs' limitations (see Section 2.6). This limitation was observed during the analysis of our dataset in the following formats:

[4]https://www.misp-project.org/taxonomies.html

Table 5. Contribution of each public taxonomy of Table 4 in the definition of the unified taxonomy.

Unified Taxonomy		Public Taxonomies																																	
Tier 1	**Tier 2**	**#Tg**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**#Tx**	**#W**
Abusive content	spam	13	1	1	1		1	3	2				1	1	1	1									10	13
Malicious Code	adware	4				1	1		1		1														4	1
	backdoor	4				2			1						1										3	1
	browser-modifier	3				2						1													2	2
	cryptominer	3	1						1						1										3	6
	dialer	4		1		2						1													3	1
	dos	14				4			1						9										3	5
	exploit	6			1	2	1		2																4	1
	hack-tool	1				1																			1	2
	misleading	8	1		1	6																			3	6
	monitoring-tool	7				2			2						3										3	8
	password-stealer	6			1				1						4										3	6
	ransomware	12	1	1	1	2	1		1		1	1			2		1								10	2
	remote-access-tool	7		1	2	2	1				1														5	1
	settings-modifier	3		1		2																			2	4
	spammer	4				1			1						2										3	2
	spoofer	2				2																			1	2
	spyware	8		1		2	2		1		1	1													6	2
	trojan	15		1		10	1				2	1													5	7
	virtool	8		1	1	2	1		1		1	1													7	3
	virus	7		1	1	2	1				1	1													6	2
	wiper	5	1						2						2										3	6
	worm	9		1	1	2	1	1	1		1	1													8	2
Information-	scanning	11	1	1			1	2	1			1							3					1	8	3
Gathering	sniffing	6		1					3			1							1						4	2
	social-engineering	17		1	1				6	4		1	1	2				1							8	12
Intrusion-	ids-alert	12		1				8				1		1		1									5	5
Attempts	brute-force	9		1				4	1			1				1			1						6	3
	unknown-exploit	3		1			1					1													3	3
	account-compromise	6		2								2				2									3	6
	system-or-application-compromise	60	4	4	1		1	7	34			2		2		1			2	2					11	6
	botnet-member	2		1								1													2	2
Availability	dos-or-ddos	24	1	3	4	4	1	1	2					1		2								5	10	6
information-	unauthorised-information-access	9		1	2			2				1				1					1	1			7	3
content-security	unauthorised-information-modification	9		1	1			3				1				1					1	1			7	3
Fraud	masquerade	6		1	1							1		1		1						1			6	2
	phishing	23	1	1	3		2	4	1	1			1	4	2	1			1		1				13	4
Vulnerable	vulnerable-service	4		1			1					1				1									4	2
Contribution of each public taxonomy in #Tags		354	12	31	23	53	18	35	66	5	9	23	3	12	27	13	1	1	8	2	3	3	5	1		
#Tier 2 categories in which public taxonomies contributed			9	25	16	20	16	10	21	2	8	21	3	7	10	11	1	1	5	1	3	3	1	1		

- *Events containing collections of IoCs from multiple attacks.* Most of these events contain IoCs with few or no correlations. For example, some of these events contain lists of malicious IPs with the primary purpose to serve as an input for a detection or prevention component. Since these events contain long lists of attributes with few to no context between each other, we opted to discard them from further analyses, not negatively impacting our results. In total, 17 events were discarded from the 1168 events.
- *Events with too many attributes.* 20% of our dataset contained events with more than 100 attributes. From the point of view of a security analyst, the more attributes an event has, the more difficult it is to analyze.

To discover the most prevalent attributes that underlie an incident category, i.e., the main threat attributes, the following analyses focused on the events with less than 100 attributes and those with too many attributes. For the latter, we intend to understand why they have so many attributes and capture which important information might be extracted from them. Thus, the following three analyses were made considering both number of attributes. These analyses combined the results by the number of attributes, aiming to differentiate the results from smaller and bigger events and consequently determine the main attributes. For this purpose, four attribute intervals were

Table 6. Unified taxonomy (excerpt of) with public taxonomy and bag of words mappings.

Unified Taxonomy		Public Taxonomies	Bag of words
Tier 1	**Tier 2**		
Abusive content	spam	cccs:email-type="spam" circl:incident-classification="spam" ecsirt:abusive-content="spam" enisa:nefarious-activity-abuse="spam" europol-event:email-flooding europol-event:spam europol-incident:abusive-content="spam" gsma-fraud:technical="spamming" information-security-indicators:iex="spm.1" maec-malware-capabilities:maec-malware-capability="email-spam" rsit:abusive-content="spam" veris:action:malware:variety="spam" veris:action:social:variety="spam"	spam, junk email, junk mail, junk e-mail, unsolicited email, unsolicited mail, unsolicited e-mail, bulk email, bulk mail, bulk e-mail, unwanted email, unwanted mail, unwanted e-mail
malware	adware	cccs:malware-category="adware" malware_classification:malware-category="adware" ms-caro-malware:malware-type="adware" veris:action:malware:variety="adware"	adware
	backdoor	maec-malware-behavior:maec-malware-behavior="install-backdoor" ms-caro-malware:malware-type="backdoor" ms-caro-malware-full:malware-type="backdoor" veris:action:malware:variety="backdoor"	backdoor
	browser-modifier	cccs:malware-category="browser-hijacker" ms-caro-malware:malware-type="broswermodifier" ms-caro-malware-full:malware-type="broswermodifier"	browser hijacker, browser modifier

considered: $I1$ - less or equal than 100, $I2$ - between 100 and 500, $I3$ - between 500 and 1000, and $I4$ - greater than 1000.

3.3.1 Distribution of Events by Attributes. This first analysis was based on the distribution of events by the four intervals of attributes. However, since we aim to get the attributes that better characterise an incident category, it was needed to determine which events are classified as an incident and which are not, distributing them along with the intervals. We resorted to the public taxonomies' tags to classify each event according to UT. More precisely, each tag from each event was compared with the public tags and, when matched, classified according to the corresponding Tier1 category of UT. The 691 tagged events in an incident category were correctly classified in UT, whereas the remaining 460 (out of 1151) were not classified because they did not have any classification tags related to incidents, so they did not match with any taxonomy. 666 of the classified events fit the first two ($I1$ and $I2$) intervals, respectively, with 550 and 116 events. It is important to note that some events were classified with more than one Tier1 category because they had more than one public tag corresponding to different UT categories.

3.3.2 Identification of Similar Attribute Types. Due to the high amount of MISP supported attribute types, a second analysis was made to identify attributes with similar types (i.e., properties) and aggregate them. For example, both MD5 and SHA1 attributes are hash values that are used as a checksum to verify data integrity, so they will be aggregated into the same group named *file hash*. By aggregating similar types of attributes, the results of the subsequent analysis will be focused on the characteristics of the attributes and not only on their type, meaning that, even if our dataset only have attributes with the type MD5, attributes with the type SHA1 will not be discarded from the results, since they belong to the same group.

3.3.3 Identification of Threat Main Attributes. This analysis had the objective of identifying the most predominant attribute groups for each Tier 1 category, based on the previous two analyses.

The four intervals of the number of attributes were considered but cumulative. This means that the first cumulative interval ($CI1$) is equal to $I1$, the second cumulative interval ($CI2$) contains all events with a number of attributes until 500, i.e., $I1$ and $I2$, and so on. Table 7 shows the results of this analysis, i.e., the most predominant attribute groups for each Tier 1 category of UT. The complete tables can be found in Appendix C [24].

As expected, the events with more attributes have a higher impact on the statistical results due to the weight of an event being directly proportional to the amount of the attributes in itself. This observation can be confirmed from the results presented in the table. As a result, when the analysis was performed over all the classified events ($CI4$ interval of attributes), some of the results had significant discrepancies compared to the analysis results restricted to events with less than 100 attributes. For example, for *information-gathering* Tier 1 category, the attribute group *network name* equals 12% of all groups when the analysis is only made over events with less than 100 attributes, and the same attribute group equals 61% of all groups when including all the classified events in the analysis ($CI4$). Since our dataset comprises events with less than 100 attributes, we have higher trust in the results gathered from those. Thus, we opted to use the result from the $CI1$ (or $I1$) interval. In a more detailed analysis on this interval for all Tier 1 categories, we noticed that four attribute groups are present in every category, namely, *Network address*, *File hash*, *Other Info*, and *File name*. Also, the attributes *URL* and *Network name* are present in all categories, except in *Vulnerable* and *information-content-security* categories. This information will be used to improve the global quality of the events by only using the most important attributes of each category.

Table 7. The most predominant attribute groups for Tier 1 categories of the unified taxonomy.

Attribute Group	$CI1$	$CI2$	$CI3$	$CI4$	Attribute Group	$CI1$	$CI2$	$CI3$	$CI4$
Abusive-content					**Malicious-code**				
URL	30%	25%	22%	17%	File hash	24%	29%	33%	32%
Network address	28%	26%	29%	25%	URL	17%	15%	13%	10%
Network name	27%	23%	20%	16%	Network address	17%	16%	15%	13%
File hash	8%	14%	15%	23%	Network name	16%	15%	13%	21%
Other Info	3%	6%	6%	8%	Other Info	15%	16%	16%	15%
File sample	2%	6%	6%	11%	File name	3%	3%	3%	2%
File name	2%	1%	1%	0%	Date	2%	2%	2%	2%
Email text	1%	0%	0%	0%	File sample	1%	2%	2%	4%
Information-gathering					**Intrusion-or-intrusion-attempts**				
Network address	35%	25%	25%	13%	Other Info	31%	23%	10%	10%
File hash	22%	23%	23%	11%	File hash	30%	31%	13%	13%
Other Info	12%	10%	10%	5%	Network name	22%	7%	6%	6%
URL	12%	12%	12%	6%	Date	7%	7%	3%	3%
Network name	12%	23%	23%	61%	File name	4%	3%	1%	1%
File name	2%	3%	3%	2%	Network address	3%	27%	54%	54%
Vulnerability	1%	0%	0%	0%	URL	3%	2%	11%	11%
Email text	1%	0%	0%	0%	Email address	1%	0%	0%	0%
Availability					**Information-content-security**				
Network name	33%	33%	33%	33%	Other Info	52%	52%	52%	52%
Network address	25%	25%	25%	25%	File name	29%	29%	29%	29%
Other Info	23%	23%	23%	23%	File hash	11%	11%	11%	11%
File hash	14%	14%	14%	14%	Date	3%	3%	3%	3%
Rule	2%	2%	2%	2%	File sample	1%	1%	1%	1%
Date	1%	1%	1%	1%	Network address	1%	1%	1%	1%
File name	1%	1%	1%	1%	Regkey	1%	1%	1%	1%
URL	1%	1%	1%	1%	URL	1%	1%	1%	1%
Fraud					**Vulnerable**				
Network name	50%	49%	58%	81%	File hash	53%	53%	53%	53%
File hash	14%	23%	13%	6%	Other Info	18%	18%	18%	18%
URL	11%	4%	5%	2%	File name	13%	13%	13%	13%
Other Info	11%	9%	11%	5%	Network name	11%	11%	11%	11%
Email address	5%	1%	3%	1%	Rule	3%	3%	3%	3%
Network address	4%	5%	3%	2%	Network address	2%	2%	2%	2%
Rule	2%	1%	0%	0%	Process other info	1%	1%	1%	1%
File name	1%	3%	2%	1%	Agent	0%	0%	0%	0%

3.4 OSINT References to External Platforms

Another key finding from our dataset was many references to external platforms in the form of links, namely 5325 links from 228 domains. More than 90% of the links pointed to VirusTotal[5], an online service that analyses files and URLs enabling the detection of viruses, worms, trojans and other kinds of malicious content using antivirus engines and website scanners. Additionally, these platforms like VirusTotal tend to provide APIs to access information without using the website interface. However, the amount of these references increases the time that an analyst requires to analyse the event since the analyst needs to jump between platforms to gather information and process it manually. We consider this as a TIP's limitation (not pinpointed on Section 2.6, neither by [13][14][44]) which can easily be turned into a benefit and it is considered in our proposed solution.

4 AUTOMATED EVENT CLASSIFICATION AND CORRELATION PLATFORM

This section presents the overall design of our proposed solution, called *Automated Event Classification and Correlation Platform* (AECCP), which aims to improve the quality threat intelligence produced by TIPs by classifying and enriching it automatically. In practice, the solution is composed of four core modules, each one focused on one or more limitations verified in our data analysis detailed in Section 3 and some of those presented in Section 2.6, and a fifth module that interconnects the other four and manages all AECCP's operations.

Table 8. Addressed limitations and correspondent proposed solutions.

ID	Limitation	Solution	Module	Section
LT10	Diverse data formats	Every event will be normalized to a standard format	Classifier	4.3
LT7	Threat knowledge management limitations	Every event will be classified according to the unified taxonomy defined in Section 3.2		
LT2	Limited technology enablement in threat triage	The classification of each event will be automated, based on its data (description of the attack, anti-virus reports, etc.)		
LT5	Limited advanced analytics capabilities and tasks automation			
LT1	Shared threat information is too voluminous	Each event will have a simplified view only containing the most predominant attributes stated in Section 3.3	Trimmer	4.4
LT3	Data Quality	Events containing links to VirusTotal will be enriched with information provided by the platform. Additionally, events containing hashes and URLs will also be enriched using the same method.	Enricher	4.5
LT8	Focus on tactical IoCs			
LT9	Trust related issues			
LT4	Limited analytics capabilities	When at least 2 events from the same category have an attribute in common, a cluster will be created in order to help an analyst identify related events and to be included in network defence mechanisms	Clusterer	4.6
LT6	Focus on data collection			
LT11	Shared intelligence without expiration date			

Regarding the limitation related to the volume of shared information, we propose an approach to reduce the number of attributes per event based on the most predominant attributes of its category, which were determined in Section 3.3. Moreover, for incident taxonomy management, we propose to classify every event according to the unified taxonomy defined in Section 3.2. Since AECCP will analyse and classify events in an automated way, it also increases technology enablement in threat triage. Furthermore, we propose a solution to enrich the data quality of an event based on OSINT from the VirusTotal platform. To increase the advanced analytics capabilities of MISP, we propose to create new events as clusters of enriched events from the same threat and with related attributes in common, after a correlation process that looks for relationships between attributes of different events. Table 8 depicts the limitations that we addressed in AECCP as well as the proposed solution for each one, the AECCP's module that comprises the solution, and the section it is presented. However, for a better understanding of the solutions, first, we present the symbolic representation

[5]https://www.virustotal.com/

of an event that is used along the sections and in Section 4.2 we give an overview of the platform, showing the workflow and interactions between the four modules.

4.1 Symbolic Representation of an Event

A TIP's event can be represented by the tuple $E_x =< d, ot, T, A, R >$, identified by x and where d is its description, $T = \{NULL|T_1...T_n\}$ the public taxonomy tags that classify it into malicious threat categories and custom tags created by security analysts, for example, to identify the event within the organization; $A = \{A_1...A_m\}$ the attributes, ranging from 1 to m, that characterize the event; $R = \{NULL|(A_i, A_j)...(A_u, A_v)\}$ the relations between attributes. For example, (A_1, A_2) represents the relation between A_1 and A_2 attributes. If the event is not yet classified and there is no relations between their attributes, $NULL$ is used to indicate such. Finally, all the other data of an event with minor relevance for this work will be compacted into the field ot.

AECCP follows this event representation, but the elements of AECCP's events are sets associated with UT, main and enriched attributes and their relations. We denote ${}^{u}E_x =< d, {}^{u}T, {}^{u}A, {}^{u}R >$ as being the resulting AECCP's event when the platform processes E_x, and we use the following nomenclature: ${}^{u}T = \{{}^{u}T_1...{}^{u}T_m\}$ is the UT tags that classify the event; ${}^{u}A = \{{}^{g}A, {}^{e}A\}$ is the set of attributes that characterize the event, which can be main threat attributes (${}^{g}A = \{{}^{g}A_1...{}^{g}A_j\}$) and enriched attributes (${}^{e}A = \{{}^{e}A_1...{}^{e}A_v\}$). A ${}^{e}A_j$ attribute is the resulting of an enrichment of a ${}^{g}A_j$ attribute, i.e., a ${}^{g}A_j$ attribute is enriched with external information from VirusTotal and with antivirus information associated with the result of VirusTotal (resulting in ${}^{e}A_j$). ${}^{u}R = R({}^{u}A)$ the relations between attributes from ${}^{u}A$. Also, we denote by ${}^{u}C_y$ the cluster resulting from the correlation and aggregation tasks performed by AECCP over ${}^{u}E$ events.

4.2 AECCP Overview

AECCP is a platform that interacts with TIPs (e.g., MISP) to generate new events with their quality threat intelligence increased. In other words, it classifies, enriches and correlates the events received by TIPs, and does all the work in an automated manner. The platform is composed of five modules – *Classifier*, *Trimmer*, *Enricher*, *Clusterer*, and *Orchestrator* – which the first four perform together all the work and the last coordinates the workflow between the other four. Figure 1 depicts the overview of its architecture and the workflow between the four modules.

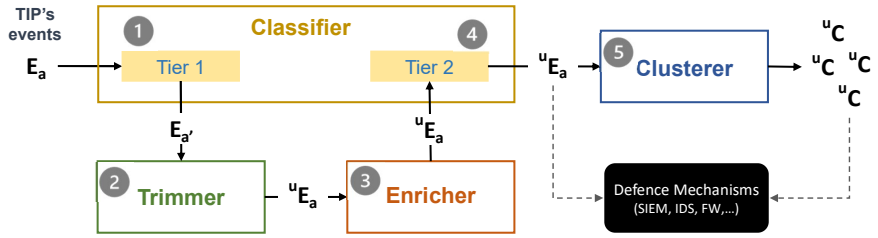

Fig. 1. Overview of the AECCP.

(1) An event E_a, from the TIP database (e.g., MISP), serves as input to the *Classifier* module, without suffering any pre-processing from TIP. The module aims at classifying each event according to UT. In order to get the most accurate classification, E_a is firstly normalized to a standard format and then is only classified according to the Tier 1 category of UT. Afterwards, the event is updated with Tier 1 tags (${}^{u}T$ tag set), transforming it into $E_{a'}$.

(2) The *Trimmer* module aims at reducing the volume of attributes in an event based on the relevancy of those attributes. The module receives $E_{a'}$, iterates over its attributes, and creates ${}^{u}E_a$, an AECCP event with the most relevant attributes ${}^{u}A_i$ and the ${}^{u}T$ tag set from $E_{a'}$.

(3) The new event (${}^{u}E_a$) is then sent to the *Enricher* module to enrich it with information from VirusTotal. In this module, ${}^{u}A$ attributes in the event containing URLs or hashes are updated with information from the VirusTotal. Additionally, the module adds an associated enriched attribute to the event for each ${}^{u}A_i$ attribute that was updated (enriched). This new attribute will support the output of antivirus engines, website scanners and analysis tools (that allowed the update). At the final, ${}^{u}E_a$ is updated with both attributes and its relationship ($R({}^{u}A)$).

(4) ${}^{u}E_a$ is now reprocessed by the *Classifier* module, but this time according to the Tier 2 category of UT. Since the event was enriched (by the *Enricher*) with information not existent at the beginning of the processing, the *Classifier* can classify the event more accurately. In this step the Tier 1 ${}^{u}T_x$ tags are updated with Tier 2 ${}^{u}T_{x.y}$ tags (e.g., $[unified :^{u} T_1 =^{u} T_{1.2}]$).

(5) The *Clusterer* module aims at creating clusters of events that share the same threat category and have at least an ${}^{u}A_i$ attribute in common. Other events that share at least one Tier 2 ${}^{u}T_{x.y}$ with ${}^{u}E_a$ and have at least one valuable attribute ${}^{u}A_i$ (attributes that provide context to a specific attack) in common with ${}^{u}E_a$ are clustered in a new cluster event ${}^{u}C_i$. Moreover, this module is recursive, meaning that it tries to find other events related to every event added to the cluster. Additionally, multiple new ${}^{u}C_i$ can be created by Clusterer if ${}^{u}E_a$ has more than one distinct Tier 2 category tag.

Both results provided by the second pass of the Classifier and the Clusterer can be integrated into defence mechanisms (e.g., firewalls, IDS, IPS, and SIEMs) installed in the organization?s IT infrastructure to protect the organization from cyber-attacks.

Figure 2 presents the detailed workflow within and between the four modules. The following four sections are dedicated to each module to describe its operation in detail.

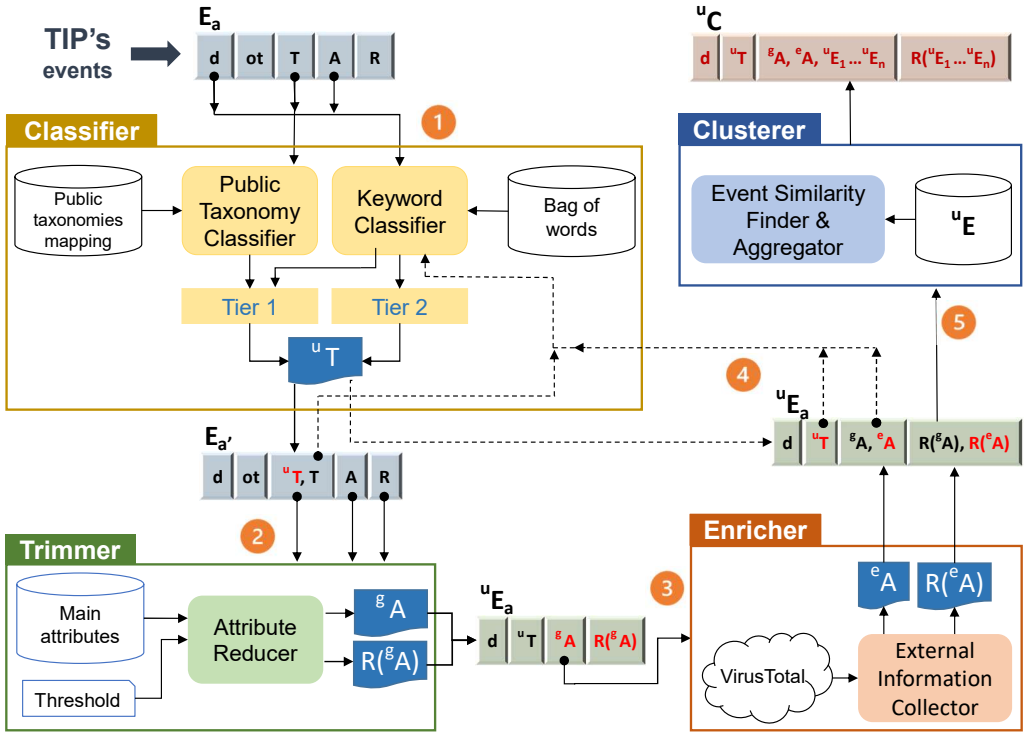

Fig. 2. The detailed workflow within and between the modules of the AECCP.

4.3 Automated Event Classification

As explained in Section 3.2, the high diversity of classification tags can be a disadvantage from the point of view of threat knowledge management (LT7). Furthermore, the diversity of data formats that OSINT can take (LT10) can have a negative impact on this management, making OSINT processing difficult. Additionally, due to this diversity, most events must be manually analysed to identify their categories and classify them as such. Since most threat triage and periodisation processes rely on the event category (LT2), this manual process creates an unwanted delay in the subsequent processes (LT3). To reduce these limitations, AECCP comprises the *Classifier* module that automatically classifies events according to the UT after they have their data format normalised and based on the tag, description and attributes information of the TIP's events. To do so, the Classifier resorts to two methods – *classification based on public taxonomies tags* and *classification based on keywords*.

Regarding the first method, the Classifier takes advantage of the mapping information from Table 6 to find every public taxonomy tag T_i to map to a UT tag ${}^{u}T_i$. In other words, each TIP's event will have its tags scanned and matched against the UT mapping table. When matched, the corresponding UT tag ${}^{u}T_i$ is added to the ${}^{u}T$ list, if not already in the list. In the end, the T tag list of the event is updated with the ${}^{u}T$ list it found. For example, if an event has two public tags related to the same threat category, e.g., the tags `[cert-xlm:information-gathering="scanner"]` and `[circl:incident-classification="scan"]`, the UT tag `[unified:information-gathering="scanning"]` will be added to the ${}^{u}T$ tag list once, and then this list will be added to the T list. Note that the ${}^{u}T$ tags follows the same scheme of tags from public taxonomies, i.e., `[taxonomy:Tier1 tag="Tier2 value"]`. We identified UT by `unified`.

For the second method, the Classifier uses the bag of words from the last column of Table 6 to identify keywords related to a UT category based on the information contained in the description, attributes and custom tags (tags that do not belong to a public taxonomy) of the TIP's events. As we previously mentioned, some events hold important details in their descriptions that can help an analyst identify the category of the incident. Moreover, it is also possible to gather important information from attributes and custom tags of an event to better classify it. Therefore, events will also have their custom tags, description and attributes scanned and matched against the bag of words. When matched, the related UT tag ${}^{u}T_i$ is added to the ${}^{u}T$ tags list, if not already in the list. Later, this list will be added to the T list. Unlike the first method, this method can classify events that were not tagged yet (i.e., without classification tags; $T = NULL$). As an example, if the word *phishing* is found in the description of an event with no public taxonomy tags, the event will be updated to contain the ${}^{u}T_i$ tag `[unified:fraud="phishing"]` in its ${}^{u}T$ list.

Each event is processed two times by the classifier module, in steps 1 and 4 of Figure 2, each time according to a different UT Tier. In step 1, the module classifies E_a according to Tier 1 and updates it with the Tier 1 ${}^{u}T$ tags it founds, resulting in thus $E_{a'}$. This step uses the two classification methods described above. On the other hand, in step 4, the Classifier updates the ${}^{u}T$ tags determined in step 1, but now according to Tier 2. It uses the *classification based on keywords* method, but now it resorts to information driven by the processing of the *Trimmer* and *Enricher* modules (see next two sections), which add information that did not belong to the initial event (E_a), respectively, the main attributes (${}^{g}A$) and the enriched attributes (${}^{e}A$). Therefore, this information is matched against the bag of words for each Tier 1 category already found, obtaining the Tier 2 associated with Tier 1. In addition, new ${}^{u}T_i$ Tier 1 can be found during the analysis if those attributes contain information that allows such. Afterwards, the Tier 1 tags from the ${}^{u}T$ list are updated with Tier 2 tags, in the form `[unified:`${}^{u}T_i$ $Tier1$ `=` ${}^{u}T_{i.j}$ $Tier2$`]` (e.g., `[unified:fraud="phishing"]`).

As final remarks, if E_a could not be classified according to Tier 1 category (in step 1) due to lack of information, the event proceeds without ${}^{u}T$ tags since the subsequent modules will enrich it; so it will receive other information. Step 4 will reprocess and classify it according to Tier 1 and Tier 2 categories. If it still could not be classified, the event exits the pipeline and is not processed by the further modules.

Algorithm 1 represents the main logic behind the Classifier, where the processing of each event is separated in Tier 1 classification (step 1, lines 1–3) and Tier 2 classification (step 4, lines 5–9) based on the state of the event that was passed into the Classifier. For each tier classification is called the function *classifyTier1* and *classifyTier2*. The *classifyTier1* function (presented in Algorithm 2) uses the *Public Taxonomy Mapping* (lines 5–8) and the *Bag of Words* (lines 9–16) for discovering the ${}^{u}T_i$ Tier 1 tags. Algorithm 3 shows the logic behind the *classifyTier2* function, which also uses the same repositories for processing the information of step 4.

Algorithm 1: Main logic overview of the *Classifier* module.

```
   input  : E_a for step 1 or uE_a for step 4;
            state indicating if Classifier is being called on step 1 or step 4
   output : E_a' for step 1 or uE_a updated for step 4
1  if state == 1 then
2      uT ← classifyTier1(E_a);
3      E_a' ← concat(E_a, uT);
4  else
5      if uT == NULL then
6          uT ← classifyTier1(uE_a);
7          uE_a ← update(uE_a, uT);
8      uT ← classifyTier2(uE_a);
9      uE_a ← update(uE_a, uT);
```

Algorithm 2: Unified taxonomy Tier 1 classification.

```
   input  : E_a for step 1 or uE_a for step 4;
            Public Taxonomies Mapping PubTaxMap;
            Bag of Words BagOfWords
   output : NULL or the uT list containing uT_i Tier 1 UT tags
1  d ← [d from E_a | d from uE_a ];
2  T ← [T from E_a | NULL];
3  A ← [A from E_a | [gA, eA] from uE_a ];
4  uT ← NULL;
5  foreach Tier 1 UT uT_i in PubTaxMap do
6      foreach public taxonomy PubTax_x related to uT_i do
7          if PubTax_x belongs to T && uT_i does not belong to uT then
8              add(uT_i, uT);
9  foreach Tier 1 UT uT_i in BagOfWords do
10     foreach word w related to uT_i do
11         if d contains w && uT_i does not belong to uT then
12             add(uT_i, uT);
13         else
14             foreach attribute att in A do
15                 if att contains w && uT_i does not belong to uT then
16                     add(uT_i, uT);
17 return uT
```

4.4 Event Simplification

The amount of shared information derived from events with too many attributes (LT1) was another limitation verified in Section 3.3. Both manual and automated analyses of events are impacted by unnecessary information. This type of information mainly acts as *good to know*, in opposite to

need to know, creating noise and consequently adding complexity to the event. To minimize this limitation, we propose the *Trimmer* module. The Trimmer automatically trims the less relevant attributes from events, based on their UT Tier 1 tags and according to the predominant attributes (i.e., *good to know* information) resulting from the analysis presented in Section 3.3.

Algorithm 3: Unified taxonomy Tier 2 classification.

```
input : Event ᵘE_a on step 4;
        Event E_a' on step 4;
        Public Taxonomies Mapping PubTaxMap;
        Bag of Words BagOfWords
output: NULL or the ᵘT list containing ᵘT_ij Tier1:Tier2 tags
T ← T from E_a';
d ← d from ᵘE_a;
A ← [ᵍA, ᵉA] from ᵘE_a;
foreach Tier 1 UT ᵘT_i in ᵘT do
    foreach Tier 2 UT ᵘT_j related to ᵘT_i in PubTaxMap do
        foreach public taxonomy PubTax_x related to ᵘT_j do
            if PubTax_x belongs to T && Tier1:Tier2 UT ᵘT_ij does not belong to ᵘT then
                add(ᵘT_ij, ᵘT);
    foreach Tier 2 UT ᵘT_j related to ᵘT_i in BagOfWords do
        foreach word w related to ᵘT_j do
            if d contains w && Tier1:Tier2 UT ᵘT_ij does not belong to ᵘT then
                add(ᵘT_ij, ᵘT);
            else
                foreach attribute att in A do
                    if att contains w && Tier1:Tier2 UT ᵘT_ij does not belong to ᵘT then
                        add(ᵘT_ij, ᵘT);
foreach Tier 1 UT ᵘT_i in ᵘT do
    remove ᵘT_i
return ᵘT
```

Each event served as an input to the module will have its attributes scanned and mapped according to the attribute groups. Afterwards, based on a global relevancy threshold defined by the security analyst for each attribute group (e.g., 10%) and the Tier 1 tags, if the attribute in analysis belongs to a group with greater relevance than the threshold and based on results of Table 7, the attribute will be marked as being a main threat attribute. For cases where the event has no Tier 1 ${}^{u}T$, it is processed the same way as if it had all Tier 1 of ${}^{u}T$ tags, thus not losing any predominant attributes. Finally, if both attributes of an event's relation were considered main threat attributes, the relation is added to the final event (i.e., to ${}^{u}E_a$). This verification and addition are made for every relation the event contains.

Summarily, the module receives $E_{a'}$ as input, identifies its main attributes and the relations between them, and then creates the ${}^{u}E_a$ event with the description of $E_{a'}$, the ${}^{u}T$ tags, the list ${}^{g}A$ of main attributes, and their relations ($R({}^{g}A)$). Algorithm 4 shows the logic behind this module, which follows the process described throughout this section.

4.5 OSINT-based Event Enrichment

As explained in Section 3.4, more than 90% of the links contained in events pointed to VirusTotal online platform. The references to external platforms increase the time an analyst requires to analyse an event since he needs to jump manually between platforms to gather information. Moreover, enriching events with additional information gathered from external sources can significantly improve other processes and tasks (LT3, LT8) if this information is related to a predominant attribute group (a main threat attribute) (LT9).

AECCP integrates an event *Enricher* module that takes advantage of the references to external platforms to enrich the quality threat intelligence of events. Hence, the module automatically enriches events containing main threat attributes with links to VirusTotal, URLs or file hashes.

Algorithm 4: Algorithm of the *Trimmer* module.

```
input : Event E_a' on step 2;
        Main Attributes MainAtts;
        Treshold tresh
output: Event ᵘE_a
d ← d from E_a';
ᵘT ← ᵘT from E_a';
A ← A from E_a';
R ← R from E_a';
ᵍA ← NULL;
R(ᵍA) ← NULL;
ᵘE_a ← new AECCP event;
add (d, ᵘE_a);
add (ᵘT, ᵘE_a);
if ᵘT == NULL then
    foreach Tier 1 UT ᵘT_i in PubTaxMap do
        foreach attribute group attG in MainAtts do
            if attG % > tresh && attG does not belong to ᵍA then
                add(attG, ᵍA);
else
    foreach Tier 1 UT ᵘT_i in ᵘT do
        foreach attribute group attG in MainAtts related to ᵘT_i do
            if attG % > tresh && attG does not belong to ᵍA then
                add(attG, ᵍA);
foreach attribute att in A do
    foreach attribute group attG in ᵍA do
        if type(att) is related to attG then
            add(att, ᵍA);
add(ᵍA, ᵘE_a);
foreach attribute relation attR in R do
    if both attributes from attR are in ᵍA then
        add(attR, R(ᵍA));
add(R(ᵍA), ᵘE_a);
return ᵘE_a
```

Algorithm 5 illustrates the data flow made by this module, which follows the process presented next. Each ${}^{u}E_a$ event processed by Enricher will have its ${}^{g}A$ main attributes scanned. If any of these attributes have any URL or file hash, it is parsed to extract them. In addition, since VirusTotal links contain IoCs in the target URL, they are also extracted by the same procedure. For each extracted IoC (URL or file hash), a request is sent to VirusTotal, and a report is received containing the most known antivirus engines, website scanners and analysis tools regarding that IoC. This information will update those ${}^{g}A_i$ attributes with URLs and file hashes, transforming them into enriched attributes, ${}^{e}A_i$. Additionally, complementary information can be received like hashes according to different hashing algorithms. Such information is also stored in ${}^{e}A_i$ attributes, and a relationship between them is created (denoted by $R({}^{e}A_i)$).

4.6 Event Clustering

Creating correlations between events is one key feature that helps SOC analysts identify threats with similarities, such as source, target, payload, threat actor, and used tools. However, as mentioned previously, most TIPs have limited advanced analytics capabilities (LT4) related to event correlation.

MISP has its built-in correlation algorithm that allows an analyst to identify events that have attributes in common. However, this algorithm relies on the values of the attributes and one key information, a flag, that specifies if that attribute can be correlated. This flag is inserted manually and, if not appropriately used, negatively impacts the correlation of events. For example, if a user adds an attribute to an event that indicates that the payload was sent over HTTP, the correlation of this attribute with attributes from other events will mostly be useless since many attacks use HTTP to send the payload. Therefore, we must know which attributes should be flagged as correlation information and why some attributes should not be flagged as such. Thus, it is crucial to managing event correlation properly. Moreover, this built-in algorithm does not use the information related to the event category, creating a relation between events without context.

Algorithm 5: Algorithm of the *Enricher* module.

input : Event ${}^{u}E_a$ on step 3;
External platform VirusTotal VT
output: Event ${}^{u}E_a$ with some of yours ${}^{g}A$ attributes enriched, the ${}^{e}A$ attributes

$fileH \leftarrow$ File hash attribute group;
$R({}^{e}A) \leftarrow$ NULL;
foreach *attribute group* attG *in* ${}^{g}A$ **do**
  if type(attG) *in* $FileH$ || type(attG) *is* URL || type(attG) *is* $link$ **then**
    $res\_vt \leftarrow$ get(attG, VT);
    attE $\leftarrow$ update(attG, res_vt);
    $av \leftarrow$ new attribute;
    update(av, *antivirus summary information*);
    $rel \leftarrow$ relation(attE,av);
    add(rel, $R({}^{e}A)$);
return ${}^{u}E_a$

The AECCP aims to improve the analytic capabilities (LT4) of TIPs, namely the event correlation capabilities, turning TIPs more than a data collector and repository (LT6). For that, it contains the *Clusterer* module for automatically creating clusters of events that share the same incident category and have at least one valuable main attribute in common (attributes that provide context to a specific attack, such as hashes). The resulting clusters are AECCP events that combine information about the same attack and which can be shared timely with external entities and used in defence mechanisms (LT11).

Hence, each event received by the Clusterer will have its main attributes scanned, looking for connections points with other events. For each scanned attribute, if its content does not add value when correlated, it will be skipped. Attributes' contents such as booleans, dates, and small sets of possible values like HTTP methods fit in this case because multiple events with no relation have them in common. A concrete example of this case is an HTTP flood attack, which is categorized on UT as `[unified:availability="dos-or-ddos"]`, and an intrusion using an unknown exploit as `[unified:intrusion-or-attempts="unknown-exploit"]`. Both events could be exploited using the HTTP GET method, but they do not correlate between them, meaning that they may even share some attribute's content (HTTP GET), but it does not imply that they are related. On the other hand, if the scanned attribute adds values when correlated, a search is made over the database of events to identify other events that contain the same attribute. If at least the event has a correlation with another event and both share a ${}^{u}T_i$ tag, a cluster is created. The resulting cluster contains the ${}^{u}T_i$ tag shared by events that compose the cluster, as well as all their attributes. Finally, all events that compose the cluster are added as attributes and, for each, relations are created with the attributes obtained from the correspondent source events.

In Figure 2 we can see the transformation of event ${}^{u}E_a$ processed by the Clusterer. When processed, attributes from ${}^{g}A$ and ${}^{e}A$ lists are scanned to identify valuable attribute (attributes that provide context to a specific attack). Being ${}^{g}A_x$ an valuable attribute, a search is made over

${}^{u}E$ events database to identify other events with ${}^{g}A_x$. Being ${}^{u}E_b$ an event that contains ${}^{g}A_x$ in common with ${}^{u}E_a$, ${}^{u}T$ tags from ${}^{u}E_a$ and ${}^{u}E_b$ are scanned in order to find at least one UT tag in common. Being ${}^{u}T_i$ a common tag for both events, the ${}^{u}C_{ab}$ cluster is created with the tag ${}^{u}T_i$. Furthermore, all the attributes from ${}^{u}E_a$ and ${}^{u}E_b$ are added to the cluster, where for those valuable attributes in common, i.e., that formed the cluster, their contents are concatenated (e.g., ${}^{g}A_x = [{}^{u}E_a({}^{g}A_x)||{}^{u}E_b({}^{g}A_x)]$). Additionally, ${}^{u}E_a$ and ${}^{u}E_b$ are also added as attributes to avoid losing the original events that generated the cluster, and relations are created between them. In Section 5.2.4 a real example is provided to better understand the Clusterer output.

Algorithm 6 shows the data flow of the Clusterer explained above. In lines 3 – 9, the algorithm searches upon events ${}^{u}E$ on the database to get other events with at least one attribute in common with event ${}^{u}E_a$.

Algorithm 6: Algorithm of the *Clusterer* module.

```
   input : Event uE_a on step 5;
           Database with uE events
   output: Cluster uC with events that characterize a same threat
 1 uC ← create cluster;
 2 foreach UT tag uT_ij in uT do
 3     eventList = [];
 4     foreach event uE_i in uE do
 5         if uT_ij in uT (uE_i) then
 6             foreach attribute att in [eA(uE_a), gA(uE_a)] do
 7                 foreach attribute attx in [eA(uE_i), gA(uE_i)] do
 8                     if att == attx && is not in [eA(uC), gA(uC)] then
 9                         add(uE_i, eventList);
10     foreach event uE_i in eventList do
11         eventAtt ← new event attribute;
12         eventAtt ← uE_i;
13         foreach attribute att in [eA (uE_i), gA (uE_i)] do
14             if att is not in [eA(uC), gA(uC)] then
15                 add(att, [eA(uC), gA(uC)]);
16                 rel ← relation(att,eventAtt);
17                 add(rel, R(uC));
18 return uC
```

4.7 Orchestrator

This module is responsible for ensuring that each event, at any time, follows a specific flow, and it is only processed by a module if the event has the required requirements (e.g., only can be enriched if it was already trimmed). Additionally, this module is responsible for checking for new events of TIPs, which were added via sharing or manually and initiating the AECCP processing for each event. In sum, the Orchestrator is responsible for the following tasks:

- *Fetch new TIP's events.* Periodically, it checks if there are new events from the selected OSINT feeds and adds them to the TIP's database.
- *Initiate processing of new TIP's events.* Periodically, it checks for events that were added since the last time AECCP processed an event.
- *Assure the correct workflow order.* It acts as a manager by sending each event to the correct next module. This module takes advantage of custom tags that are only used by it, and these tags store the current state of the event regarding the AECCP processing order.
- *Resume the process.* If the processing of an event is interrupted, the module can resume the processing of that event without impacting the event database by falling back to the previous event state.

4.8 Implementation

We implemented the AECCP using Python 3.7 and over the MISP. For that, AECCP resorts PyMISP[6], a Python library to access the MISP platform via their REST API. Implementing AECCP leverages built-in PyMISP functionalities to search, add or update events and attributes.

AECCP implements the five modules described in Section 4. Its modules can be considered smaller solutions and, therefore, can work regardless of each other. Also, the platform has the capability of exporting its events (i.e., ^{u}E events and ^{u}C clusters) to be used by external entities, for example, SIEMs, CSIRTS, and SOCs.

5 EVALUATION

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

(1) Is AECCP able to classify events that are not initially tagged?
(2) Is AECCP able to reclassify events previously tagged with a known incident classification taxonomy?
(3) Does AECCP simplify event triage?
(4) Is Trimmer able to reduce the number of attributes of events without losing valuable information for their classification?
(5) Does Enricher improve the quality of the events?
(6) Is AECCP able to correlate different events (threats) that share the same IoC?
(7) Is AECCP more effective than PURE and ETIP platforms?

We validated and evaluated AECCP with three datasets of events. For validation we used as ground truth the dataset we analyzed in Section 3 (Section 5.1), whereas for evaluation we used two datasets that we did not have any knowledge about their events and being one of them constituted by events generated by PURE [3] (Sections 5.2 and 5.3). Also, Section 5.3 presents an evaluation of AECCP with PURE and ETIP platforms.

5.1 Validation with the Ground Truth Dataset

In order to validate the AECCP, we used as ground truth dataset the 1,168 events we analysed in Section 3. The dataset comprises 2 totally untagged events and 1,166 tagged events, of which, from the latter, 691 events are tagged into an incident category, but several of them have multiple overlapping classification tags from different public taxonomies. The remaining 475 events are not tagged into an incident category; hence, we consider them untagged. Summing up, the ground truth contains 691 tagged events and 477 untagged events. The tagged events will serve to validate the *classification based on public taxonomies tags* method, whereas the untagged events will validate the *classification based on keywords* method, both methods from the Classifier module (see Section 4.3). However, note that we want to classify events for both UT tiers, meaning that the Classifier, Trimmer and Enricher modules will be used and validated, and the Classifier will be executed twice.

Processing the 691 tagged events with AECCP, we verified that they were correctly classified into incident categories of UT for both Tier 1 and Tier 2. The resulting classification was checked based on the manual classification we made in the data analysis section (see Section 3). Table 9, second column, shows these events classified through the eight Tier 1 categories of UT. Notice that an event can fit into different Tier 1 categories.

For the 477 untagged events, when Classifier processed them the first time, the *classification based on keywords* method was able to classify 453 of them into Tier 1 categories of UT, based on their descriptions and attribute values. The other 24 remained untagged events, carried on to the Trimmer and Enricher modules, and then re-evaluated by Classifier. We observed after this processing that

[6]https://pymisp.readthedocs.io/

16 of them were enriched with external data, but the external data only allowed to tag 8 of them in an incident category, i.e., with UT Tier 1 and Tier 2 tags. Curiously, the 2 totally untagged events were between these 8 events. For all 461 classified events, we manually inspected their information before and after they were processed by AECCP and verified that AECCP correctly tagged them. For the 16 events that the platform failed to classify, we also inspected them to find out why. We checked that they did not provide enough information in their descriptions and attributes to permit them to be associated with an incident category. In addition, the attributes that Enricher enriched did not bring valuable information that would allow their classification. The last column of Table 9 presents the 461 events classified into the eight Tier 1 categories.

Table 9. The ground truth dataset classified by AECCP over the Tier 1 incident categories of UT.

Tier1	Tagged events	Unttagged events
Abusive content	145	99
Malicious Code	607	408
Information Gathering	63	55
Intrusion Attempts	37	43
Availability	5	10
information-content-security	2	12
Fraud	34	40
Vulnerable	3	5
Total	896	672

Most of the events were classified into the *Malicious code* (malware) and *Abusive content* Tier 1 incident categories of UT, reflecting well the number of cyberattacks that have been made over the Internet. As a result, we can conclude that AECCP has a precision[7] of 1 (i.e., 100%) when classifies events previously labelled by public taxonomies. In contrast, AECCP, when processes untagged events, its precision depends on the information that their descriptions, attributes and external data can provide about the threats they report. Based on our ground truth, from the 477 untagged events, the platform correctly classified 461 (TP) and did not have false positives (FP, events classified wrongly into incident categories), meaning thus it had a precision of 1. However, since it was not able to classify 16 out of the 477 events, we consider these events as being false negatives (FN), and so it had a false negative rate of 0.033 and a recall[8] of 0.966. Overall, based on the 1,168 events, AECCP classified 1,152 (without false positives) and missed 16. Thus, it had a precision of 1, a recall of 0.986, a false negative rate of 0.013, and a F1-Score[9] of 0.992.

We measured the time that AECCP takes to process both types of events (tagged and untagged). This time is strongly related to the quantity of data included in the events and that the platform has to analyze, which depends on diverse factors, namely the number of the public taxonomy tags, the number of attributes, and the amount of external data. As expected, the greater the amount of data, the longer it takes to process it. Also, tagged events take longer than untagged events, considering that both types of events have the same number of attributes and the same amount of external data. It is explained by the fact that the former have their tags analyzed by the *classification based on public taxonomies tags* method, while the latter does not. For the tagged events with less than 100 attributes, the average time for processing an event by AECCP is 30 seconds. Considering all 691 tagged events, it takes an average of 41 seconds to consume an event, with a standard deviation (Std) of 17 seconds, which means that, at most, it takes approximately one minute to process an event. Regarding untagged events, the processing times are shorter, namely: (i) 24 seconds on average for events with less than 100 attributes; (ii) 31-seconds average for processing any event

[7]Precision = $TP/(TP + FP)$
[8]Recall = $TP/(TP + FN)$
[9]F1-Score = $2 * (Precision * Recall/(Precision + Recall))$

out of the 477 events, with an 11-seconds std; (iii) a maximum of 42 seconds to process an event. Therefore, the maximum time AECCP takes to process an event is one minute. Although it seems a bit long, we consider it acceptable given that it is the cost of reducing to zero the time spent by SOC analysts in analyzing and classifying events, which might incur classification errors.

5.2 Processing Dataset of MISP's Events

This section assesses the ability of AECCP to process a dataset composed of 64 MISP's events that were not previously processed by the platform. The following sections present the characterization of the dataset and its processing by AECCP's modules.

5.2.1 Dataset characterization. The dataset's events were provided from different providers – CIRCL, CUDESO, inThreat, VK-Intel, ESET and MalwareMustDie – where 54 of the events were from the first two sources. From the 64 events, approximately 77% (49 events) of them did not contain any tags related to a known incident classification taxonomy, meaning that those events were not yet classified. These events will serve to evaluate the AECCP ability to classify events with the *classification based on keywords* method and to answer question 1. Regarding the volume of attributes of the events and distributing them according to the same four intervals used in Section 3.3, the dataset is mainly composed of events with less than 100 attributes, 90% of the 64 events.

To get a detailed evaluation of our solution, we choose to perform a more in-depth analysis of the (remaining) 15 events that, contrarily to the others, 49 events, were initially classified with a known incident classification taxonomy. We choose these events since they can be used to evaluate almost all use cases that AECCP deals with, except the AECCP ability to classify events that are not initially classified, which can be evaluated by comparing the number of unclassified events initially and after being processed by AECCP. Table 10 shows a more detailed view of the tags and the attributes of these 15 events, namely, their public taxonomy tags (column 2), the total number of tags (TT, column 3), including tags that did not add information about the type of the threat (e.g., Traffic Light Protocol), the number of classification tags related to threat incidents (CT, column 4), and the number of attributes (Att, column 5). As we can observe, all of the events have more tags than those that really classify events with known incidents, having some of them a considerable number of tags not associated with incidents, such as events 1, 11, 12. As we already stated, such tags do not add value of threats, making the SOC analyst waste time analyzing irrelevant information.

5.2.2 Event classification. This section looks to evaluate AECCP ability to classify events into UT for Tier 1 and Tier 2. So, the Classifier module will be evaluated for all its functionalities, but also the Trimmer and Enricher modules since these two modules support the Classifier in the classification of events. Also, this section aims to answer the first three questions.

After AECCP processed the dataset, 61 out of the 64 events were classified, increasing 72% of the number of classified events. We recall that only 15 events were initially classified with public taxonomy tags. Only 3 (out of the 64) events were not classified into UT due to the lack of information in their descriptions and the absence of indicators that the Enricher could process (e.g., URL), thus adding more information to the events helpful the Classifier. The classification was verified manually, meaning that AECCP correctly processed all events.

The 49 out of the 64 events without any tags related to a known incident classification taxonomy were processed only using the *classification based on keywords* method. AECCP was able to classify 46 of them, meaning that the 3 events that were not classified belong to this data subset. Overall, 75% (46) of 61 classified events by AECCP were classified only based on keywords, meaning that AECCP can classify events that are not initially classified, answering positively to question 1.

Regarding the analysis targeted to the 15 events initially classified with a known incident classification taxonomy, the platform was able to use both classification methods and classify

Table 10. Characterization of dataset of MISP's events and results of processing of it by AECCP.

	MISP's events				AECCP				
E_x	**Public taxonomy tags**	**TT**	**CT**	**Att**	**Unified taxonomy tags**	**TT**	**CT**	**AT**	**AE**
1	circl:incident-classification="spam"	12	1	17	malicious-code="virus" malicious-code="worm" malicious-code="spammer" abusive-content="spam"	4	4	13	13
2	enisa:nefarious-activity-abuse="spear-phishing-attacks"	4	1	84	fraud="phishing"	1	1	78	92
3	malware_classification:malware-category="Botnet"	4	1	10	availability="dos-or-ddos" malicious-code="exploit" malicious-code="dos" malicious-code="backdoor" malicious-code="remote-access-tool" malicious-code="cryptominer"	6	6	10	10
4	malware_classification:malware-category="Ransomware"	5	1	18	vulnerable="vulnerable-service" malicious-code="exploit" malicious-code="ransomware"	3	3	18	42
5	malware_classification:malware-category="Ransomware"	3	1	9	malicious-code="wiper" malicious-code="ransomware"	2	2	8	8
6	circl:incident-classification="malware" malware_classification:malware-category="Downloader" malware_classification:malware-category="Rootkit" malware_classification:malware-category="Botnet"	8	4	73	malicious-code="virtool" malicious-code="cryptominer" malicious-code="trojan" malicious-code="remote-access-tool"	4	4	43	53
7	malware_classification:malware-category="Ransomware"	5	1	7	malicious-code="ransomware"	1	1	7	7
8	circl:incident-classification="malware"	8	1	29	malicious-code="virus" malicious-code="trojan"	2	2	29	36
9	circl:incident-classification="malware"	4	1	11	malicious-code="trojan"	1	1	11	11
10	enisa:nefarious-activity-abuse="spear-phishing-attacks"	8	1	115	fraud="phishing"	1	1	105	173
11	ecsirt:intrusions="backdoor" veris:action:malware:variety="Backdoor" ms-caro-malware:malware-type="Backdoor" ms-caro-malware-full:malware-type="Backdoor"	38	4	17	malicious-code="virtool" malicious-code="trojan" malicious-code="backdoor" fraud="phishing"	4	4	15	34
12	ms-caro-malware:malware-type="Trojan" ms-caro-malware-full:malware-type="Trojan" ecsirt:malicious-code="trojan" CERT-XLM:malicious-code="trojan-malware" malware_classification:malware-category="Trojan"	10	5	10	malicious-code="trojan"	1	1	10	10
13	ecsirt:intrusions="backdoor" veris:action:malware:variety="Backdoor" ms-caro-malware:malware-type="Backdoor" ms-caro-malware-full:malware-type="Backdoor"	10	4	34	malicious-code="virtool" malicious-code="backdoor" malicious-code="virus" malicious-code="cryptominer"	4	4	34	34
14	circl:incident-classification="malware" ecsirt:malicious-code="malware"	12	2	86	malicious-code="trojan"	1	1	86	86
15	ecsirt:malicious-code="trojan"	7	1	27	malicious-code="trojan"	1	1	27	166

them correctly. Almost every event was classified with a new type of threat that was not initially considered in the public taxonomy tags. For example, event E_1 from Table 10 was identified only as *spam* before being processed by AECCP, but after being processed by AECCP it was also classified as *malicious code with virus*, *worm* and *spammer*, meaning that AECCP is able to reclassify events, and so answering question 2. The sixth column of Table 10 shows the transformation of the tags of the 15 events face to their original classification presented in the second column.

From the 15 events, on average, each had 5 more tags than before processed by AECCP, increasing thus their tags from 2 to 7 (columns 4 and 8). As explained in Sections 3.2 and 4.3, AECCP classifies events according to UT and, also, based on information contained in their description, meaning that each event classification can be improved. These assumptions can increase the number of tags per event. In addition, it is important to note that, after being processed by AECCP, all of the tags on the events tag list are classification tags, contrary to before being processed by AECCP where most of the tags were not classification tags, but added information about its source and its sharing (e.g., TLP). Table 10, on columns 4 and 8, shows the number of tags regarding known incident classification taxonomy, before and after being processed by AECCP.

From the 15 events, 14 of them had their total number of tags significantly reduced (columns 3 and 7) due to two factors. The first is when an event has overlapping classification tags in its initial tag list (e.g., `[cccs:malware-category="ransomware"]`, `[cert-xlm:malicious-code="ransomware"]`) since they are transformed into a UT tag after being processed by AECCP. The second one is when an event has non-classification tags in its initial tag list (e.g., TLP) since they are removed after being processed by AECCP. From the point of view of a SOC analyst, the exclusion of non-classification tags and the inclusion of new classification tags based on OSINT can simplify event triage since all the tags in the event tag list add value to the analyses, answering thus to question 3.

5.2.3 Attribute trimming and enrichement. This section looks to evaluate AECCP ability to trim and enrich events. More precisely, we evaluated the Trimmer and Enricher modules and sought to answer the fourth and fifth questions.

Before being processed by AECCP, our dataset had approximately 90% of the events with less than 100 attributes. After being processed by AECCP, the number of events with less than 100 attributes decreased to 85% of the initial number. This means, at first glance, that our solution enriches more than it trims, adding more attributes than removing.

To understand this overall attribute increment, we analysed the number of attributes of the events in three specific phases: before being processed by the Trimmer, exactly after being processed by the Trimmer and, finally, after being processed by the Enricher. From the results of this analysis, we can see that, on average, the Trimmer removes 12 attributes per event, and the Enricher adds 54 attributes per event, thus increasing 44 attributes per event. Enricher's increase is because it can add a maximum of 6 new attributes for each hash and 12 new attributes for each URL. For example, if an event has attributes containing 3 hashes and 3 URLs, the Enricher will add 54 attributes to the event. Summing up, on average, the number of attributes in the three phases is 49, 37, and 91. Therefore, the attribute increment is due to the Enricher, which overlaps the Trimmer's effect since this last trims the event attributes effectively.

Similar to the Classifier evaluation, we also evaluated the Trimmer and Enricher impact on the 15 events. Table 10 shows the number of attributes on the three phases, namely, before they are processed by Trimmer and Enricher (Att, column 5), after Trimmer (AT, column 9) and after Enricher (AE, last column). We verified that AECCP could reduce the number of attributes of some events depending on the type of attributes of those events, so Trimmer, in these cases, reduced the number of attributes effectively. This was observed in 6 out of the 15 events. On the other hand, we also verified that those events that their attributes contain hashes and URLs, their number of attributes was increased by Enricher. Summing up, 7 events were increased, where 4 were first trimmed. Two of the remaining 8 events were trimmed but not enriched, and the other 6 were neither trimmed nor enriched. Overall, 6 had their number of attributes increased, 3 had their attributes reduced, and the remaining 6 maintained their number of attributes.

We evaluated with and without these two modules to answer the fourth and fifth questions. Table 11 shows the results of this evaluation, where compares the number of classification tags of the 15 events whether they did not pass through the Trimmer and the Enricher (columns 2, 6 and 10), with the number of classification tags whether they only did not pass through the Enricher (columns 3, 7 and 11) and with the number of classification tags when processed by all modules (columns 4, 8 and 12). As we can observe, all the events have the same number of tags in columns 2–3, 6–7 and 10–11, meaning that the Trimmer does not remove valuable information for the classification of events, answering positively to question 4. We can also observe from columns 4, 8 and 12 that the number of classification tags of 4 events were increased (E_3, E_8, E_9, and E_{15}), where 2 of them leveraged from the enrichment provided by Enricher (E_8, and E_{15}). Therefore, we conclude that the Enricher improved the quality of the events, answering question 5.

Table 11. Trimmer and Enricher impact on the number of tags of the 15 events.

E_x	without T & E	with T	with T & E	E_x	without T & E	with T	with T & E	E_x	without T & E	with T	with T & E
1	4	4	4	6	4	4	4	11	4	4	4
2	1	1	1	7	1	1	1	12	1	1	1
3	5	5	6	8	1	1	2	13	4	4	4
4	3	3	3	9	0	0	1	14	1	1	1
5	2	2	2	10	1	1	1	15	0	0	1

5.2.4 Clustering. This section aims to assess AECCP ability to correlate different events that share mutual IoCs, i.e., the Clusterer module, and answer the sixth question.

Since our evaluation dataset is small (64 events) and, therefore, Clusterer might not create many clusters, we allowed these events to be correlated with events from our ground truth dataset, thus totalling 1232 events. With this approach, we were able to create 24 clusters. Table 12 details some of these clusters while the remaining are omitted since they have the same properties, except their taxonomies, as one of the clusters in this table. For example, clusters 100, 101 and 102 have exactly the same attributes and correlations, but they were created with different taxonomies (`[unified:malicious-code="worm"]`, `[unified:malicious-code="backdoor"]` and `[unified:malicious-code="trojan"]`) due to the logic behind of the Clusterer module.

Figure 3 presents one of the clusters that were created by AECCP, identified with ID 21 in Table 12. This cluster is formed by two events (1518 and 1520) that have a common attribute, a link, and a common UT tag, `[unified:malicious-code="ransomware"]`. The attribute in common is a link to `https:\\bleepingcomputer.com` with news related to ransomware `LockerGoga`, meaning that both events are related to the same threat. Because these two events have different information, except for the single shared link, they complement each other. This type of event correlation can be precious to a SOC analyst since he can easily gather more information about an event based on previously received events and give him more indicators that can be used in block rules and other types of defences, answering thus to question 6.

Table 12. Clusters created by the AECCP.

${}^{u}C_x$	# events	Taxonomy and Description	# Att	Mutual IoCs
1	2	malicious-code="worm" -Soft Cell case indicators -Malware with Ties to SunOrcal	416	www.tashdqdxp.com
9	3	malicious-code="trojan" -FIN7 JScript Loader Malware -APT28 XTunnel Backdoor -Turla Kazuar RAT	68	https://twitter.com/VK_Intel/status/1128079463785349121
10	2	malicious-code="virus" -FIN7 JScript Loader Malware -APT28 XTunnel Backdoor	47	https://twitter.com/VK_Intel/status/1128079463785349121
11	2	malicious-code="ransomware" -Sodinokibi ransomware -Ransomware exploits WebLogic vulnerability	69	All except one
14	2	malicious-code="cryptominer" -Botnet Malware Exploits CVE-2019-3396 -SystemTen (ELF trojan, miner, bot and rootkit)	65	CVE-2019-3396
119	2	malicious-code="backdloor" -Operation ShadowHammer -Operation ShadowHammer	53	All except three
21	2	malicious-code="ransomware" -The Norsk Hydro ransomware attack -New LockerGoga Ransomware in Altran Attack	28	https://www.bleepingcomputer.com/news/security/new-lockergoga-ransomware-allegedly-used-in-altran-attack/

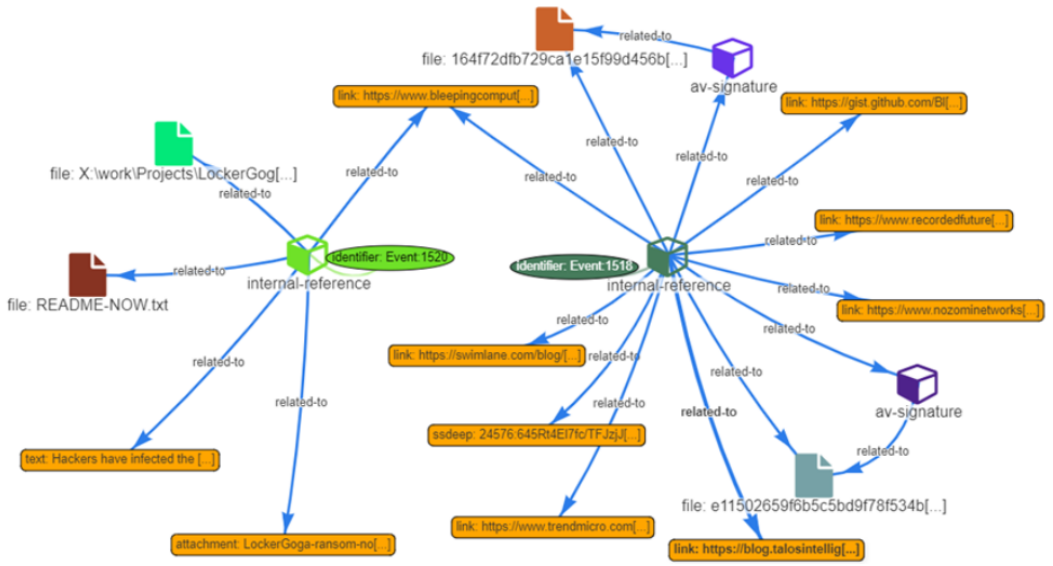

Fig. 3. Cluster 21 created by AECCP and composed of 2 events: 1518 on the right and 1520 on the left.

5.3 Processing Events with PURE and ETIP Platforms

To demonstrate the AECCP ability to process events processed by other platforms existent in literature, without losing relevant information by trimming event attributes and enriching the information they carried and, hence, their threat impact, we processed 6 events from PURE [3]. Also, we compare the resulting events with the PURE versions by submitting them to ETIP [15] to calculate the threat score (TS) of the threat value they carried.

Table 13 shows the characterization of the 6 events of PURE, namely, for each eIoC, the number of events it aggregates (#E, column 2), its description (column 3), the number of attributes it contains (#att, column 4), and its threat score measured by ETIP (TS, column 5).

The 6 events received from PURE were processed by AECCP, producing the results shown in columns 6 to 8 of the table. As we can observe, AECCP could process events from an external platform. All of the events, which were not initially tagged, were classified by AECCP (column 8). Also, the initial number of attributes (column #att) was slightly reduced (column #AT) by Trimmer. However, as explained in Section 5.2.3, AECCP adds on average 44 attributes per event when it enriches events. This increase can be seen in column #AE, a price to pay for the added value. On the other hand, this increase allowed events to gain more information, which apparently is relevant since their threat impact grew and was reflected in their TS value (last column).

Based on these results, we can answer positively to question 7, meaning AECCP improves the quality TI better than the other two platforms. Notice that the ETIP platform calculates the TS of events (enriched IoC), meaning that the platform contains an enricher module that aggregates and correlates events before calculating TS. Therefore, if the TS value of AECCP's events is higher than ETIP's events, this means that AECCP generates events with better quality than ETIP. The same is concluded about PURE.

6 IMPROVEMENTS AND FUTURE WORK

The prevention and detection of cyber-attacks have deserved significant attention from organizations, which have been adopting new strategies and defence mechanisms to protect themselves. TI has emerged as an ally of organizations, allowing them to access information about threats

Table 13. PURE events characterization, processed by AECCP, and threat score calculation by ETIP.

PURE and ETIP					AECCP and ETIP			
ID	**#E**	**Description**	**#att**	**TS**	**#AT**	**#AE**	**Unified Taxonomy**	**TS**
E1	2	- OSINT Aveo Malware Family Targets Japanese Speaking - Pivot on whois registrant 844148030@qq.com	82	1.29	77	87	malicious-code="backdloor" malicious-code="trojan"	1.29
E2	2	- OSINT - Packrat: Seven Years of a South American Threat Actor - Packrat: Seven Years of a South American Threat Actor	267	2.54	257	423	availability="dos-or-ddos" fraud="phishing" malicious-code="backdloor" malicious-code="dos" malicious-code="ransomware" malicious-code="trojan" malicious-code="worm"	2.68
E3	2	- Expansion on 596552@qq.com - New Variant of Gh0st Malware by Palo Alto Networks Unit 42	274	3.22	273	401	malicious-code="backdloor" malicious-code="trojan"	3.50
E4	3	- Spear Phishing Attack Using Cobalt Strike Against Financial Institutions - RTF files for Hancitor utilize exploit for CVE-2017-11882 - Targeted Attack in the Middle East by APT34, using CVE-2017-11882	85	2.53	78	159	abusive-content="spam" fraud="phishing" malicious-code="exploit" malicious-code="spammer" malicious-code="trojan" vulnerable="vulnerable-service"	2.58
E5	3	- EPS Processing Zero-Days Exploited by Multiple Threat Actors - Malicious Documents Targeting Security Professionals - APT28 Targets Hospitality Sector, Presents Threat to Travelers	156	2.87	146	361	information-gathering="scanning" malicious-code="backdloor" malicious-code="exploit" malicious-code="ransomware" malicious-code="trojan" malicious-code="worm" vulnerable="vulnerable-service"	3.12
E6	4	- Sakula Malware Family - Cyber-Kraken (Threat Group 3390 / Emissary Panda) - Korean Website Installs Banking Malware - Sakula Reloaded	842	3.11	821	2907	information-gathering="scanning" malicious-code="backdloor" malicious-code="trojan"	3.40

#E:number of events; #att: number of attributes; TS: threat score;
#AT: number of attributes after Trimmer; #AE: number of attributes after Enricher

that have occurred. They use TI for various purposes, namely, to verify whether their assets are vulnerable to an attack that has occurred, to update their defence mechanisms with rules and patterns on announced threats, and to check whether their assets have been victims of an attack.

TI must be timeless for organizations to be proactive on time and avoid severe damages. However, TI only announces attacks after they have already occurred, thus being a reactive notification [41] [51] and not much useful for victim organizations. To develop proactive TI, it is necessary to obtain data from the online hacker community to understand what is happening in that community and try to predict possible malicious actions. One way to do this is to access underground forums where, for example, hackers exchange technical mechanisms and tutorials of malicious tools that they can use to carry out attacks [41]. These tools can be found and purchased within the Dark-web (DW), more precisely in Dark-net markets. Also, dark-net forums are placed within the Dark-web for hacker community [2]. By accessing the DW data and collecting and analyzing it, it is possible to identify emerging hacker threats, so proactive TI [42].

The AECCP was designed in light of traditional TI, meaning that the unified taxonomy and the main threat attributes were defined based on public taxonomies and security events of traditional TI. The AECCP can benefit from DW data in various ways, namely,

- the unified taxonomy can be extended with Tier 2 tags and bag of words based on terms only observed in DW and that are related to an incident category (Tier 1 level) of UT;
- process data provided by DW sources, classifying it with the extended UT and aggregating it with (i) some other DW data associated with the same attack intent. In this case, SOC analysts can get insights about malicious actions and anticipate potential attacks that have been planned, and then be proactive and make decisions to prevent them against the organization; (ii) traditional TI that already exists from some announced misbehaviour, but no associations

and have been passed unnoticed for security analysts (e.g., some attacks that have been planned but not yet fully executed). In this case, the SOC analyst can also be proactive and activate the necessary protections against the attack; (iii) traditional TI from an already occurred attack. In this case, the resulting information is reactive, but the analyst can have access to information about the attack plan, and, from there, make some decision based on that;

- make the necessary modifications in the AECCP to accept the different formats that the DW data can be provided.

7 CONCLUSION

In this paper, we proposed and presented the Automated Event Classification and Correlation Platform (AECCP), an implementation of an approach to improve quality threat intelligence produced by threat intelligence platforms (TIPs) by classifying and enriching it automatically. AECCP is composed of a set of smaller solutions; each one focused on one or more limitations of TIPs, which were verified in a detailed data analysis over an intelligence dataset of more than 1000 security events. Regarding threat knowledge management limitations and technology enablement in threat triage limitations, the platform integrates a Classifier that classifies each event according to a *single unified taxonomy* proposed by us. To deal with the high volume of shared threat information, we proposed a Trimmer for trimming the low-value information from each event, based on *main threat attributes* we discovered upon the data analysis. AECCP contains an Enricher for data improvement that enriches each event based on intelligence collected from VirusTotal. Lastly, to address advanced analytics limitations, we proposed a Clusterer that creates clusters of events that share information and context about the same threat and represents each cluster as an AECCP event.

To prove the applicability and feasibility of AECCP, the platform was developed based on the MISP platform. AECCP was validated over more than a thousand events and tested against a dataset of 64 newer and not used events and 6 events produced by a different platform, the PURE. From these tests, we created 24 clusters, classified, trimmed and enriched by AECCP, and we were able to trim and enrich the events produced by PURE. Also, these events were processed by another platform, ETIP, to calculate their threat score. The results showed that AECCP produces quality TI better than the others platforms.

ACKNOWLEDGMENTS

This work was partially supported by the ITEA3 European through the XIVT project (I3C4-17039/FEDER-039238), the national funds through FCT with reference to SEAL project (PTDC/CCI-INF/29058/2017, LISBOA-01-0145-FEDER-029058, POCI-01-0145-FEDER-029058), and LASIGE Research Unit (UIDB/00408/2020 and UIDP/00408/2020).

REFERENCES

[1] Fernando Alves, Aurélien Bettini, Pedro M. Ferreira, and Alysson Bessani. 2021. Processing tweets for cybersecurity threat awareness. *Information Systems* 95 (2021), 101586.

[2] Nolan Arnold, Mohammadreza Ebrahimi, Ning Zhang, Ben Lazarine, Mark Patton, and Sagar Samtani. 2019. Dark-Net Ecosystem Cyber-Threat Intelligence (CTI) Tool. 92–97.

[3] Rui Azevedo, Ibéria Medeiros, and Alysson Bessani. 2019. PURE: Generating Quality Threat Intelligence by Clustering and Correlating OSINT. In *In Proceedings of the 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (TrustCom)*. 483–490.

[4] Matt Bromiley. 2016. Threat Intelligence: What It Is, and How to Use It Effectively. https://www.sans.org/reading-room/whitepapers/threathunting/paper/37282.

[5] Ping Chen, Lieven Desmet, and Christophe Huygens. 2014. A Study on Advanced Persistent Threats. In *Proceedings of the 15th IFIP International Conference on Communications and Multimedia Security*. 63–72.

[6] CIRCL.lu. 2018. CIRCL Taxonomy - Schemes of Classification in Incident Response and Detection. https://www.circl.lu/pub/taxonomy/.
[7] A. Cormack, X. Jansen, A. Moens, and P. Peters. 2015. Incident Classification / Incident Taxonomy according to eCSIRT.net - adapted. https://www.trusted-introducer.org/Incident-Classification-Taxonomy.pdf.
[8] CSIRTG. 2020. The Fastest Way to Consume Threat Intelligence. https://csirtgadgets.com/collective-intelligence-framework.
[9] Darknet. 2020. OpenIOC - Sharing Threat Intelligence. https://www.darknet.org.uk/2016/06/openioc-sharing-threat-intelligence/.
[10] Alessandra de Melo e Silva, João José Costa Gondim, Robson de Oliveira Albuquerque, and Luis Javier García-Villalba. 2020. A Methodology to Evaluate Standards and Platforms within Cyber Threat Intelligence. *Future Internet* 12, 6 (2020), 108.
[11] Quirine Eijkman and Daan Weggemans. 2013. Open Source Intelligence and Privacy Dilemmas: Is it Time to Reassess State Accountability? *Security and Human Rights* 4 (Apr 2013).
[12] ENISA. 2015. *Standards and tools for exchange and processing of actionable information.* Technical Report.
[13] ENISA. 2017. *Exploring the opportunities and limitations of current Threat Intelligence Platforms.* Technical Report.
[14] European Comission. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). https://eur-lex.europa.eu/eli/reg/2016/679/oj.
[15] Mario Faiella, Gustavo Gonzalez-Granadillo, Ibéria Medeiros, Rui Azevedo, and Susana Gonzalez-Zarzosa. 2019. Enriching Threat Intelligence Platforms Capabilities. In *In Proceedings of the 16th International Conference on Security and Cryptography, Prague, Czech Republic (SECRYPT)*. 37–48.
[16] FireEye. 2013. *Taking a Lean-Forward Approach to Combat Today's Cyber Attacks.* Technical Report.
[17] Michael Glassman and Min Ju Kang. 2012. Intelligence in the internet age: The emergence and evolution of Open Source Intelligence (OSINT). *Computers in Human Behavior* 28 (Mar 2012), 673–682.
[18] 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, Visualisation and Sharing Capabilities. *Journal of Information Security and Applications* 58 (May 2021), 102715.
[19] Gasper Hribar, Iztok Podbregar, and Teodora Ivanusa. 2014. OSINT: A "Grey Zone"? *International Journal of Intelligence and Counterintelligence* 27 (05 2014).
[20] Brian Kime. 2016. *Threat Intelligence: Planning and Direction.* paper. SANS Institute – InfoSec Reading Room.
[21] Robert M. Lee. 2020. *2020 SANS Cyber Threat Intelligence (CTI) Survey.* paper. SANS Institute – InfoSec Reading Room.
[22] Jerome Leonard. 2020. TheHive Project: Open Source, Free and Scalable Cyber Threat Intelligence & Security Incident Response Solutions. https://blog.thehive-project.org/tag/soltra-edge/.
[23] Martin E. Dempsey. 2013. *Joint Intelligence (JP 2-0).* Technical Report.
[24] Cláudio Martins and Ibéria Medeiros. 2020. Additional info on the paper submitted to ACM TOPS. https://sites.google.com/view/siteaddinfo-tops.
[25] Troy Mattern, John Felker, Randy Borum, and George Bamford. 2014. Operational Levels of Cyber Intelligence. *International Journal of Intelligence and Counterintelligence* 27 (12 2014).
[26] Amanda Mckeon. 2016. Reduce Business Risk With an Effective Threat Intelligence Capability. https://www.recordedfuture.com/threat-intelligence-capability/.
[27] Microsoft. 2018. Security intelligence. https://docs.microsoft.com/en-us/windows/security/threat-protection/intelligence/.
[28] Jelena Mirkovic and Peter Reiher. 2004. A taxonomy of DDoS attack and DDoS Defense mechanisms. *ACM SIGCOMM Computer Communication Review* 34 (May 2004).
[29] MISP. 2020. MISP Taxonomies. https://www.misp-project.org/datamodels/#misp-taxonomies.
[30] MISP. 2020. Open Source Threat Intelligence Platform & Open Standards For Threat Information Sharing. http://www.misp-project.org.
[31] MITRE. 2020. CRITs: Collaborative Research into Threats. https://crits.github.io/.
[32] OASIS. 2020. Introduction to STIX. https://oasis-open.github.io/cti-documentation/stix/intro.html.
[33] OASIS. 2020. Introduction to TAXII. https://oasis-open.github.io/cti-documentation/taxii/intro.html.
[34] Bank of England. 2016. Understanding Cyber Threat Intelligence Operations. https://www.bankofengland.co.uk/-/media/boe/files/financial-stability/financial-sector-continuity/understanding-cyber-threat-intelligence-operations.pdf.
[35] Kris Oosthoek and Christian Doerr. 2020. Cyber Threat Intelligence: A Product Without a Process? *International Journal of Intelligence and CounterIntelligence* 0, 0 (2020), 1–16.
[36] J. Pastor-Galindo, P. Nespoli, F. Gómez Mármol, and G. Martínez Pérez. 2020. The Not Yet Exploited Goldmine of OSINT: Opportunities, Open Challenges and Future Trends. *IEEE Access* 8 (2020), 10282–10304.

[37] Andrew Ramsdale, Stavros Shiaeles, and Nicholas Kolokotronis. 2020. A comparative analysis of cyber-threat intelligence sources, formats and languages. *Electronics* 9, 5 (May 2020).
[38] Robert M. Lee. 2016. Intelligence Defined and its Impact on Cyber Threat Intelligence. https://www.robertmlee.org/intelligence-defined-and-its-impact-on-cyber-threat-intelligence/.
[39] C. Sabottke, O. Suciu, and T. Dumitras. 2015. Vulnerability disclosure in the age of social media: exploiting twitter for predicting real-world exploits. Proceedings of the 24th USENIX Security Symposium, 1041–1056.
[40] A. Saini, Manoj Gaur, and Vijay Laxmi. 2014. A taxonomy of browser attacks. (Jan 2014), 291–313.
[41] Sagar Samtani, Ryan Chinn, Hsinchun Chen, and Jay F. Nunamaker Jr. 2017. Exploring Emerging Hacker Assets and Key Hackers for Proactive Cyber Threat Intelligence. *Journal of Management Information Systems* 34, 4 (2017), 1023–1053.
[42] Sagar Samtani, Hongyi Zhu, and Hsinchun Chen. 2020. Proactively Identifying Emerging Hacker Threats from the Dark Web: A Diachronic Graph Embedding Framework (D-GEF). *ACM Transactions on Privacy and Security* 23, 4 (Aug 2020).
[43] C. Sauerwein, C. Sillaber, Andrea Mussmann, and R. Breu. 2017. Threat Intelligence Sharing Platforms: An Exploratory Study of Software Vendors and Research Perspectives. *Wirtschaftsinformatik und Angewandte Informatik* (2017).
[44] SWIFT. 2019. The Evolving Cyber Threat to the global banking community. https://www.swift.com/pt/node/147646.
[45] Symantec World Headquarters. 2011. *Advanced Persistent Threats: A Symantec Perspective.* Technical Report.
[46] ThreatConnect. 2019. *Threat Intelligence Platforms. Everything You've Ever Wanted to Know But Didn't Know to Ask.* ThreatConnect.
[47] Wiem Tounsi. 2019. *What is Cyber Threat Intelligence and How is it Evolving?* John Wiley & Sons, Ltd, Chapter 1, 1–49.
[48] Wiem Tounsi and Helmi Rais. 2018. A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Computers & Security* 72 (Jan 2018), 212–233.
[49] Cynthia Wagner, Alexandre Dulaunoy, Gérard Wagener, and Andras Iklody. 2016. MISP: The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform. In *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security.* 49?56.
[50] Webroot. 2014. *Threat Intelligence: What is it, and How Can it Protect You from Today's Advanced Cyber-Attacks.* Technical Report. Gartner.
[51] Ryan Williams, Sagar Samtani, Mark Patton, and Hsinchun Chen. 2018. Incremental hacker forum exploit collection and classification for proactive cyber threat intelligence: An exploratory study. In *2018 IEEE International Conference on Intelligence and Security Informatics.* 94–99.

Cláudio Martins is a Cybersecurity analyst belonging to the Cyber Threat Unit at Banco Santander. He holds a Master in Information Security from the Faculty of Sciences of the University of Lisbon. He participated in the H2020 project DiSIEM in activities related to threat intelligence and security data analytic platforms. His research interests include cybersecurity, SIEM environments and threat intelligence.

Ibéria Medeiros received the Ph.D. degree in computer science at the Faculty of Sciences of the University of Lisboa. She is currently an Assistant Professor with the Faculty of Sciences, Department of Informatics, University of Lisbon (FCUL), Lisboa, Portugal. She is the author of software security and cybersecurity tools, among which WAP (Web Application Protection) is the most known, and an OWASP project. Currently, she is the Principal Investigator of the SEAL national project and the XIVT European project and has been involved in international and national projects, including the ADMORPH, DiSIEM, SEGRID, and MASSIF European projects, and the REDBOOK national project. Her research interests include software security, cybersecurity, vulnerability and attack detection, and machine learning. Prof. Medeiros is a LASIGE Research Unit and the Navigators Research Group member.