www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 16(3), May-Aug 2025, 441-457. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2025.v16i3.848

THREATMOSAIC: collection, curation and enrichment of indicators of compromise (IOCs)

Sergio Mauricio Martínez Monterrubio[1], Juan Frausto Solís[2,], Juan Antonio Recio García [3]*

[1] School of Engineering, Universidad Internacional de La Rioja (UNIR), 26006 Logroño, Spain. sergiomauricio.martinez@unir.net

[2] Graduate Program Division, Tecnológico Nacional de México/Instituto Tecnológico de Ciudad Madero, Ciudad Madero 89440, Mexico. juan.frausto@itcm.edu.mx.

[3] Group of Artificial Intelligence Applications, Department of Software Engineering and Artificial Intelligence, Faculty of Computer Science, Universidad Complutense de Madrid, Ciudad Universitaria, 28040 Madrid, Spain. Juan A. Recio-García (jareciog@fdi.ucm.es).
*Corresponding author: Juan Frausto Solís (juan.frausto@gmail.com).

Abstract. In this research experimental software is developed for the classification, analysis and enrichment of indicators of compromise (IOCs), codenamed THREATMOSAIC. This software can import IOCs in bulk, classifying them according to whether they are IPv4, IPv6, URLs, MACs, e-mails, DNS domains or MD5, SHA1 and SHA256 hashes, sorting and sanitizing them in an effective and efficient way. All this is combined with the STIX2.1 standard, generating a directional graph enriched with information obtained from analysis through third-party REST APIs. Mainly information collected through services such as Virus Total, Abuse IPDB, IP Stack or Whois. Finally, the software allows sharing threat information in STIX2.1 format through the TAXII protocol via a server to which requests can be made from threat exchange platforms.
Keywords: threat indicators of compromise, cyber intelligence, threat information sharing IOC.

Article Info
Received February 01, 2025
Accepted March 21, 2025

1 Introduction

Today, many organizations, both public and private, have seen an increase in the frequency with which they suffer cyber-attacks that result in security incidents affecting their assets [1]. A worrying reality that makes cyber intelligence a tool in the fight against cybercrime. For this reason, sharing information about what, how and even who has attacked an organization becomes an essential task that helps to prevent other organizations from falling victim to incidents of the same nature in the immediate future [4]. Nowadays, intelligence sharing platforms, and specifically IOCs, are an invaluable source of information whose widespread use is of vital importance for information protection and security. An IOC is the description of a cybersecurity incident, activity and/or malicious artefact through patterns to be identified in a network or endpoint, thus improving incident management capabilities [5]. Although it may seem obvious that they should be properly categorized, ordered and structured, the lack of homogeneity and standardization of data makes it difficult to get the most out of IOCs. There are countless security services that add a multitude of new IOCs generated from successful discoveries and detections in a variety of situations and locations. A circumstance that makes all this information useless from a qualitative point of view, due, in many cases, to its shortcomings, errors or lack of elaboration. In fact, only 11.2% of CSIRTs update policies and rules based on IOC findings and lessons learned in an automated way [6]. CSIRT stands for Computer Security Incident Response Team. A CSIRT is a specialized team responsible for identifying, managing, and mitigating cybersecurity incidents within an organization or network. These teams play a crucial role in incident response, threat intelligence, and cybersecurity defense, ensuring swift action against security breaches, malware attacks, and other cyber threats.

This means that much of the work they perform is mainly executed manually when obtaining information from IOCs. In this research, an experimental software called THREATMOSAIC was developed to facilitate the task of processing and analyzing IOCs effectively, efficiently and at low cost before sharing them on intelligence-sharing platforms by automating certain processes. To do this, based on lists of IOCs shared in the ThreatFeed repository, data enrichment and information discovery techniques were applied using third-party REST APIs, such as Virus Total or URLScan [7][8]. In addition, to provide homogeneity and standardization of the enriched IOCs, the findings obtained have been combined with the STIX2.1 standard format and the

TAXII protocol when sharing the information [9] [10]. Finally, three case studies were carried out to evaluate and compare the results with four other existing applications on the market: IOCExtractor, Maltiverse, Anomali STAXX and Eleven Paths TheTHE, successfully determining the great capabilities of THREATMOSAIC when classifying, sorting and massively extracting information from lists of unstructured IOCs.

2 Related Work

The losses caused by these cybercrimes both at the end of 2019 and throughout 2020 amounted to around 1% of global GDP, or close to one trillion US dollars or, as shown in Figure 1, more than 800 billion euros [2][3]. Cyber threat intelligence, or cyber intelligence, is the result of the confluence of four fundamental pieces where, in one way or another, attackers, targets, infrastructure and capabilities are interconnected [11]. The information around these four concepts is used for intelligence preparation of battlefield (IPB), a term widely used in the military [12]. The objective is to gain knowledge of the adversary or attacker, knowing his motivations and objectives, represented in the socio-political axis of the diamond model in Figure 2, and knowing his techniques, capabilities and infrastructure, represented in the technical axis of the figure. The goal is to establish the best defence and counterattack against a threat by means of appropriate methods, procedures and tools to detect and mitigate it. This model should be able to answer who carries out the attack, what means, or infrastructure has been used to carry it out, what is the specific category of attack and, above all, how sophisticated is the attack methodology perpetrated by the attackers.

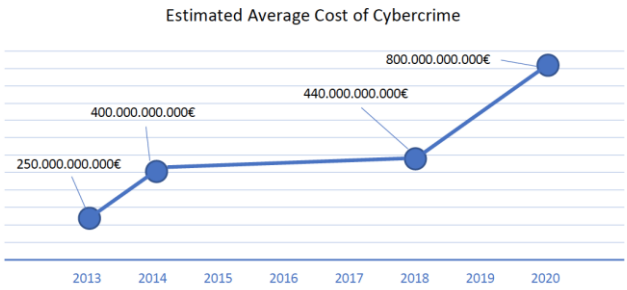

Figure 1. Estimated Average Cost of Cybercrime. Source: The Hidden Costs of Cybercrime, McAfee [2].

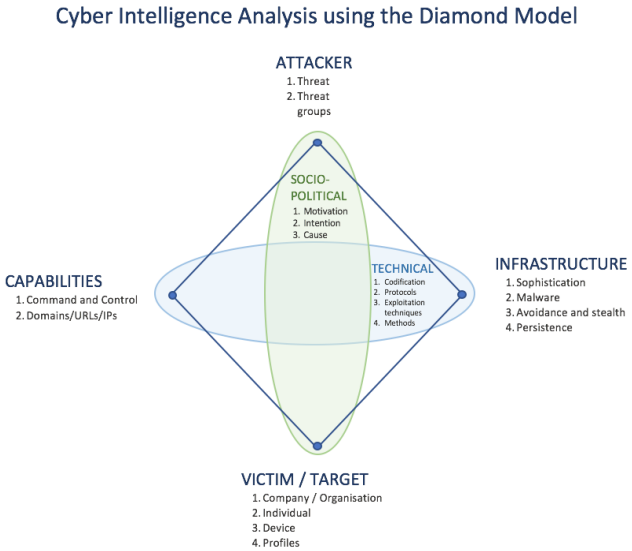

Figure 2. Diamond Model. Source: own elaboration based on [13].

In other words, the outcome of this type of intelligence is directly linked to the level of maturity in threat detection. In general, it is difficult to identify relevant data about the attacker's behaviour, beyond atomic indicators. That is, if we take the Detection Maturity Level (DML) model as a reference model, most organisations are at detection maturity level 1 on a scale from 0 to 8, as can be seen in Figure 3 [14]. This scale is directly linked to both the robustness and accuracy of detections. This means that the lower the level on this scale, the more inaccurate the detections and the less robust the detection maturity. To increase the maturity of threat detection, the companies Check Point, Cisco, Fortinet and Palo Alto Networks founded the Cyber Threat Alliance (CTA), which has large partner companies such as Anomaly, Avast, Eleven Paths, Symantec and Panda, among many others [15]. CTA members have made it their mission to ensure a safer digital ecosystem through collaboration and cyber intelligence sharing.

Through their expertise, infrastructure and commitment, they have built the foundation to counter the malicious actions of cybercriminals. This is where cyber intelligence sharing platforms become important. The aim of these platforms is to automate such intelligence sharing and in turn to enlarge the threat intelligence sharing community. However, in most cases, within these large cyber intelligence sharing platforms, the information shared tends to be unprocessed and untreated, with CSIRTs responsible for post-processing. A fact that in most cases becomes an arduous and difficult task as a result of the large volumes of data that are shared and that in many cases there is a tendency to use this information without even verifying its veracity, accepting it as good, updated and contemporary information because it is often provided by large companies positioned in the sector [16]. There are an infinite number of platforms with diverse characteristics that make the market wide and competitive, where information is massively injected every day of the year at all hours, but which sometimes makes the task of the CSIRTs difficult due to lack of time and resources to cross-check the aforementioned information [17-25]. As noted above, one of the most widely shared elements for characterising threats are IOCs. Thanks to IOCs it is possible to identify malicious activity in a network, detect data breaches, malware infections or similar events [26].

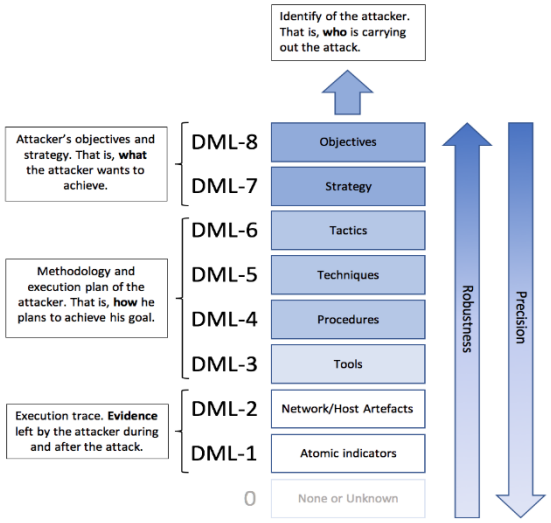

Figure 3. DML Model. Source: own elaboration based on [15].

They jeopardise the confidentiality, integrity and availability of information. So, if security teams are monitoring these indicators, they can anticipate events. The most frequently shared types of IOCs are calculated and atomic indicators, linked to tactical intelligence. They are easy to identify or replace by both responders and attackers, respectively. However, it is not so easy for attackers to modify their behaviour, i.e. their tactics, techniques and procedures (TTP). This is why behavioural indicators, associated with operational intelligence, are the IOCs that can provide the most information when dealing with an attacker [27]. This limits the risk within an organisation, as can be seen in the Pyramid of Pain in Figure 4, which depicts the relationship between the types of indicators that can be used to detect an attacker's activities and the damage they can cause when they have been identified [28].

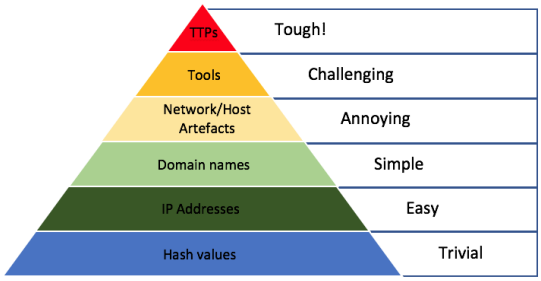

Figure 4. The Pyramid of Pain. Source: own elaboration based on [29].

When it comes to characterising, the information provided by IOCs, standardization is extremely important. Some examples are the IODEF, OpenIOC, STIX or MAEC standards [30-33]. Thanks to these standards we can process the shared data in different formats such as XML, CSV or TXT files that many cybersecurity researchers use to share their findings in different repositories such as IOC Bucket or ThreatFeeds [34,35]. It is precisely for this purpose of processing and obtaining information from IOCs that applications such as Anomali STAXX, ElevenPaths TheTHE, Maltiverse, Cortex, IOC Extractor or OpenCTI have emerged as tools for cybersecurity analysts [36-41]. The techniques for enriching IOCs can be divided into three main blocks: scanning, detonation and reversing. As NATO's Cooperative Cyber Defence Centre of Excellence (CCDCOE) points out in its malware reverse engineering manual, scanning techniques fall into the category of static malware analysis techniques. Among the most commonly used tools and techniques are web-based scanning such as Virus Total, string analysis, the PEiD tool that provides compilation information of malicious executable files, the CFF Explorer tool that allows extracting the compilation date of the malicious file and the type of architecture, and others such as Resource Hacker and PeStudio [42]. Techniques such as Passive DNS Replication or Whois stand out. Passive DNS Replication is the process of capturing DNS queries and responses in real time, using this data to build partial replicas of as many DNS zones as possible [43]. Authors such as Marco Balduzzi, Leyla Bilge, Christopher Kruegel and Engin Kirda have developed a system called EXPOSURE that uses large-scale passive DNS analysis techniques to detect domains involved in malicious activity [44]. The second is a public directory, which before the entry into force of the GDPR made it possible to find out who owned a DNS domain or IP address and other relevant information. In 2016, Masahiro Kuyuma, Yoshio Kakizaki and Sasaki Ryoichi proposed a method for detecting malicious activity using the information provided by this directory query [45]. Thanks to the Command-and-Control Domains (C&C Domains) and the rest of the domains, information is obtained through Whois, mainly the characteristics of the domain added at the time of registration. With all these characteristics, a training model is generated using machine learning techniques that allow malicious domains to be detected.

In the manual, the CCDCOE includes this type of sandboxing techniques within those intended for dynamic malware analysis, where techniques are also used to study and analyse the behaviour of malware (malware behaviour analysis) on a controlled computer using applications such as Process Monitor and Process Explorer from Microsoft SysInternals, Regshot or INetSim. Within this type of dynamic analysis, there are also debuggers, like a disassembler, but with their own particular characteristics, capable of determining a breakpoint by analyzing strings, symbols and intermodular calls, obfuscation or patching to avoid possible reverse engineering. Among these techniques related to sandboxing, some standout such as Cuckoo Sandbox, where several researches have allowed to generate enrichment in indicators of compromise in an automated way [46-48], URLScan which is a free service popularly used for web site scanning and analysis, Microsoft Sandbox or ANY.RUN [49].
Lastly, malware reversing techniques aim to study malicious code to identify the vulnerabilities it exploits when it infects a system, its attack vectors, the level of infection and the measures that can be implemented to protect against it [50]. Reversing is sometimes confused with reverse engineering. However, they are different concepts. As with the previous techniques of scanning and detonation, reversing is just another piece of the malware reverse engineering puzzle. It can bring enrichment to an IOC. A successful example of this type of technique is the Dridex Analysis Toolkit developed by AppGate Labs, which is capable of automatically extracting IOCs from Dridex banking malware. Specifically, this software implemented in Python allows extracting the ID of the botnet with which the malware communicates, extracting the C&C IP addresses, decrypting character strings and network communications, generating an enrichment from the initial IOC [51].

Therefore, what was proposed in this research is the development of software that addresses the underlying problem considering the aforementioned. To this end, the Python programming language was chosen for programming the backend, taking advantage

of its capabilities when it comes to discovering knowledge [52] and the Flask framework. In the case of the Frontend, lacking advanced knowledge of JavaScript, a study of the most popular and easy to learn languages was carried out based on the data provided by Stack Overflow [53], the decision was made to use the React JS programming language because of its versatility and lower learning curve compared to other competitors such as Angular JS or Vue.js. For the database, the characteristics of SQLite and MongoDB were compared, and it was determined that the use of MongoDB would facilitate software development thanks to its compatibility with JSON, the format used in the current version of the STIX standard. Finally, the possibility of uploading the resulting software to a cloud platform was considered. Gartner's magic quadrant was used to determine the leading services in this respect, with the giants Google Cloud Platform, Microsoft Azure and Amazon Web Services standing out [54]. The most cost-competitive solution was sought [55] and Amazon Web Services was chosen as it better adapted to the needs of the project. However, this point has not yet materialized.

3 Objectives and methodology

The overall objective of this research is mainly based on an experimental software development that facilitates CSIRTs to process, analyse and share IOCs effectively, efficiently and at low cost before sharing them on intelligence sharing platforms. The specific objectives are divided into three phases and are as follows:

1) Clean-up and management phase. This consists of obtaining IOCs and locating overlaps, gaps and errors during their recording. Performing an appropriate sorting and classification, detecting possible collisions and compiling the information of interest necessary to improve the information contained in the IOC itself.
2) Analysis and enrichment phase. It takes advantage of the fact that certain IOCs contain elements that can be analysed by scanning, detonation or reversing using third-party REST APIs. In such a way that the results of this analysis are added to the content of the IOC itself as an enrichment and improvement of its informative value.
3) Re-injection phase. During this phase, the IOC is introduced into a repository accessible by a threat exchange platform with the set of improvements made, both in its characterisation/classification and in the information obtained through third-party REST APIs or using malware analysis techniques.

Scrum was used for the planning and development of this research, incorporating some design principles and good software safety practices during the secure software development life cycle (S-SDLC). Figure 5 shows the six sprints carried out divided into three phases: sanitisation and tidying, analysis and enrichment, and re-injection.

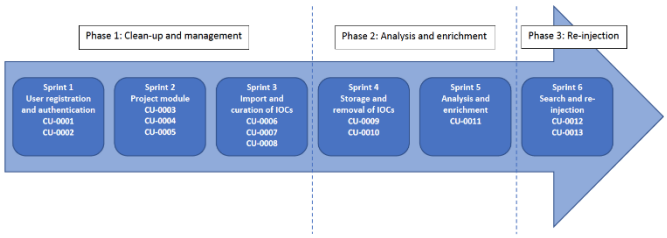

Figure 5. Phases and sprints of software development. Source: Own elaboration.

4 THREATMOSAIC

The specifications of the STIX2.1 standard and the TAXII2.1 threat exchange protocol are detailed below. The software requirements and each of the implementation phases of the THREATMOSAIC application are also specified.

4.1 Analysis of STIX2.1 Standard Specification

STIX (acronym for Structured Threat Information Expression) is a standardised language for describing and exchanging cyber threat intelligence. It is designed to be shared via various protocols such as TAXII through cyber intelligence platforms such as MISP. STIX is structured to describe threats based on the motivations, skills and capabilities of cyber criminals [56]. That is, it is a language designed to describe the Tactics, Techniques and Procedures (TTPs) of attackers. Its development was led by MITRE, with the collaboration of the National Cybersecurity & Communications Integration Center (NCCIC), the US Department of Homeland Security and US-CERT. Although version 1.0 initially used the XML6 format, version 2.1 is now based on JSON7 as the main serialisation implementation. STIX is published under an open-source licence, is compatible with multiple threat exchange protocols and has extensions that allow other standards to be included, such as MAEC, CAPEC, IODEF, VERIS, OVAL, OpenIOC or Snort and YARA rules. Another of its advantages is that it allows for a visual representation that provides cybersecurity analysts with a highly readable and easy-to-understand tool capable of describing the most complex threats, Figure 6.

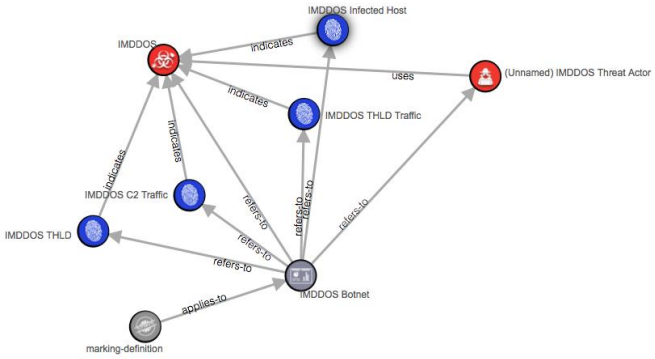

Figure 6. Example of a graphical representation of a cyber threat in STIX2.1: OASIS.

When planning the design of THREATMOSAIC, the peculiarities of STIX2.1 were considered. On the one hand, as can be seen in Figure 6, STIX uses objects to describe a cyber threat to represent it by means of a directed graph. Specifically, STIX has different types of objects: 18 STIX Domain Objects (SDOs), which are the nodes of the graph, and 2 STIX Relationship Objects (SROs), which are the ones that allow the SDOs to be related to each other, i.e. the edges of the graph. In addition, it also has STIX Cyber-observable objects (SCOs) that represent atomic indicators such as an IP address, a DNS domain name or a URL, among many others. In relation to SROs, it is important to bear in mind that the STIX2.1 standard includes in its appendix B, a summary table of the relationships, where the resource, the type of relationship and the objective are established. STIX2.1 provides us with all the sufficient elements to represent all the elements that make up the intelligence linked to a threat in a visual way in the form of a graph. In addition, it also provides us with all the necessary vocabulary to characterise these elements. Therefore, the possibilities provided by this perspective of representing and relating the different elements of a threat to generate and share IOCs is extremely versatile and dynamic. All this, if we consider that this representation arises from a JSON code.

4.2 Analysis of TAXII2.1 Protocol Specification

Now that we have visualised how IOCs can be generated in STIX2.1 format, it is time to explain how to share these IOCs using a secure protocol. That is, the TAXII2.1 protocol. Trusted Automated Exchange of Intelligence Information is a protocol used at the application layer designed to facilitate the exchange of intelligence in a secure way thanks to the HTTPS protocol. Designed to be used in combination with STIX, TAXII is compatible with OpenIOC or CybOx. Its operation is defined as follows:

4.2.1 Use of a REST API to interact with the Collection and Channel services based on a Client/Server architecture.
4.2.2 Two services:

1) Collection: consists of a service where TAXII clients and servers exchange intelligence through a request/response model.
2) Channel: consists of a service where TAXII clients exchange intelligence with other TAXII clients through a

publish/subscribe model.

For this research, due to time constraints, use was made of the experimental TAXII server written in Python, medallion, to generate a hypothetical repository where threat intelligence could be shared with exchange platforms.

4.3 Software Requirements Analysis

In the following, starting from the basis established by the IEEE-STD 830-1998: Software Requirements Specifications [57] and taking as a reference, the good practices and guidelines described therein, the fundamental aspects of the Software Requirements Specification (SRS) document are detailed. Specifically, thanks to the requirements specification, a total of 13 use cases were identified, summarised in Figure 7. The diagram mentions CU-00number. The abbreviation "CU" stands for "use cases". The diagram illustrates different roles (User, Administrator, and Guest) along with their corresponding use cases, which refer to specific interactions or functionalities that a user role can perform within a system.
Each role has a defined set of use cases:

- **User:** Basic functionalities such as registration, authentication, and project creation.
- **Administrator:** Advanced management capabilities, including project selection, deletion, importing and handling Indicators of Compromise (IOCs), and data analysis.
- **Guest:** Limited access with permissions to search and re-inject IOCs.

This structure clarifies the hierarchy of access and actions within the system.

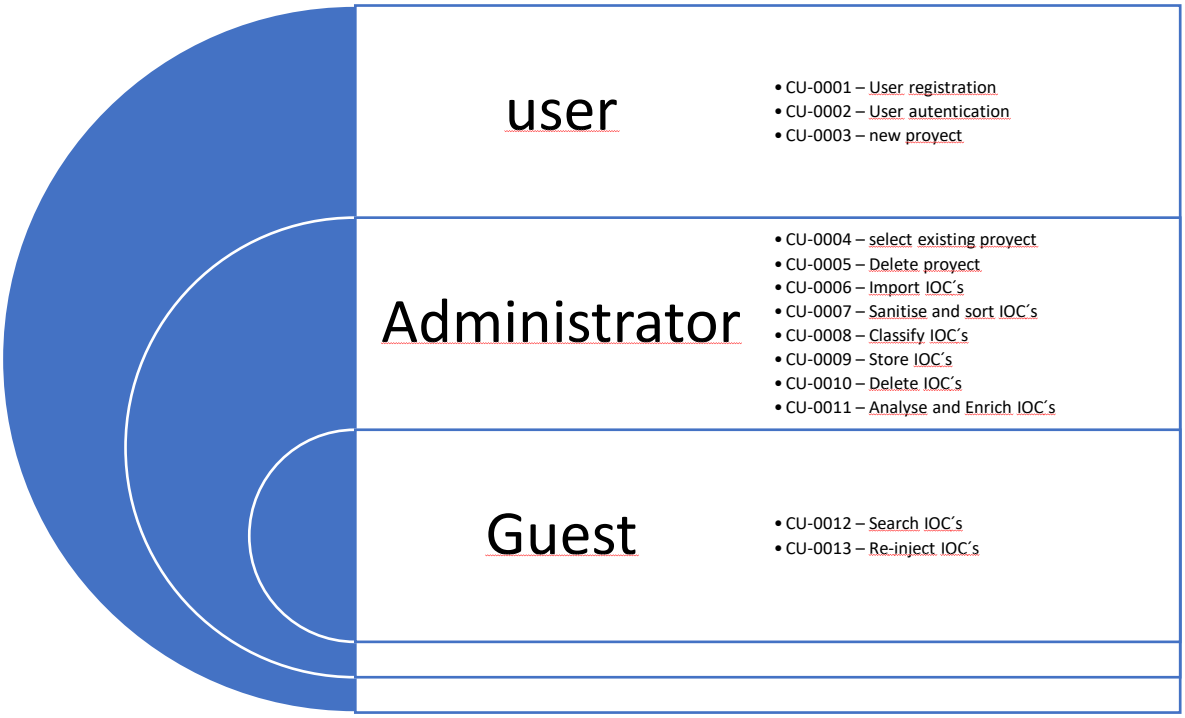

Figure 7. Use Case Diagram. Source: Own elaboration.

Each of these was addressed in the different phases of development described below and depicted in Figure 5 on the previous page. As for the domain model, a close look at Figure 8 identifies the entities, the attributes it possesses and the relationships between the different elements. It should be noted that the user entity is not used to authenticate the user in the application, since it was decided for design reasons to take advantage of the integration of Firebase in React JS. This domain model was used to build the database in Mongo DB, taking advantage of the versatility offered by a NoSQL database. Finally, the architecture diagram in Figure 8 shows that THREATMOSAIC is a web application based on a layered architecture: first is the presentation

layer, followed by the business logic layer and, finally, the data layer. Likewise, the Frontend is shown, where user requests are received and forwarded to the Backend, which communicates with the different REST APIs that provide enrichment to the IOCs imported into the application. At the same time, it is the Backend that communicates with both the database and the TAXII server to re-inject the indicators of compromise. In addition, the TAXII server stores the information in the MongoDB database.

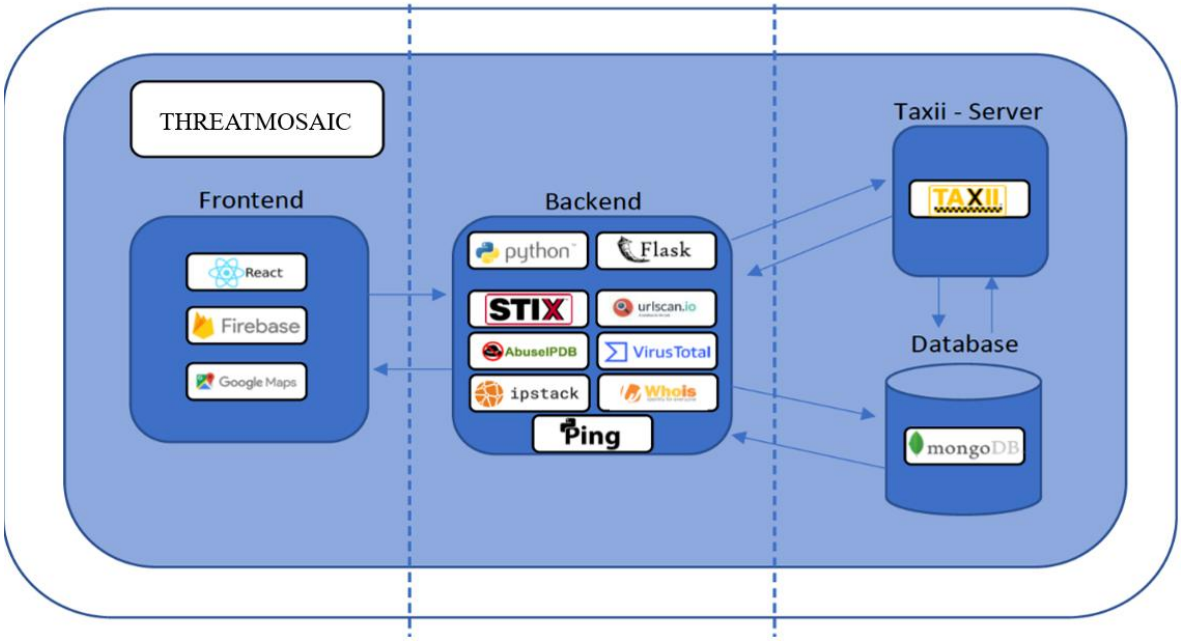

Figure 8. Architecture of THREATMOSAIC. Source: Own elaboration.

4.4 Phases of Software Development

After the software specifications and requirements analysis phase, following the methodology described in section III. Objectives and Methodology, we proceeded to software development.

4.4.1 Phase 1: Clean-up and management

This phase was divided into three sprints. Sprint 1 focused on two Use Cases, CU-0001 and CU-0002 referring to user registration and authentication. Taking advantage of the Frontend implementation using React JS we chose to use Firebase to implement this functionality [58]. This allows the minimum privilege to be established at a security level for the different functionalities depending on the role of each user. In sprint 2, the projects module, Use Cases, CU-0003, CU-0004 and CU-0005 were developed. For this, the logic of linking and storing projects based on a user attribute was followed. It is possible to access a saved project, change its name or delete it. The first time a project is opened or if it has no IOC stored, the IOC extractor, a fundamental tool of the application, is opened. In addition, the project module allows you to remove IOCs from an analysis or add new ones. Phase 1 ends with sprint 3 which corresponds to the import and curation of IOCs, Use Cases, CU-0006, CU-0007 and CU-0008. Specifically, we took advantage of the Python library based on an IOC extractor, called extract_iocs, to create a module that allows to massively extract IOCs both from text pasted in the application and from files with *.txt or *.json extension. We started from the library developed by the GitHub user Moses Shwartz [59] applying some improvements and solutions to problems that the developer had encountered, adapting it to the reality of this research. One of the improvements introduced in this module was to fix the code so that it also detects, classifies and returns IOCs based on URLs, MAC addresses and IPv6 addresses, shown in Table 1. In addition, the code was modified to provide for the possibility of returning IOCs directly in STIX2.1 format.

Table 1. Regex used to identify and classify IOCs based on URLs, MAC addresses and IPv6 with THREATMOSAIC. Source: own

Regex used to identify and classify IOCs with THREATMOSAIC	
URL	\b[a-z]{3,}\:\/\/(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),]|(?:%%[0-9a-fA-F][0-9a-fA-F]))+\b
MAC	([0-9a-fA-F]{2}[:|-]){5}([0-9a-fA-F]{2})
IPv6	(([0-9a-fA-F]{1,4}:){7,7}[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,7}:|([0-9a-fA-F]{1,4}:){1,6}:[0-9a-fA-F]{1,4}|([0-9a-fA-F]{1,4}:){1,5}(:[0-9a-fA-F]{1,4}){1,2}|([0-9a-fA-F]{1,4}:){1,4}(:[0-9a-fA-F]{1,4}){1,3}|([0-9a-fA-F]{1,4}:){1,3}(:[0-9a-fA-F]{1,4}){1,4}|([0-9a-fA-F]{1,4}:){1,2}(:[0-9a-fA-F]{1,4}){1,5}|[0-9a-fA-F]{1,4}:((:[0-9a-fA-F]{1,4}){1,6})|:((:[0-9a-fA-F]{1,4}){1,7}|:)|fe80:(:[0-9a-fA-F]{0,4}){0,4}%%[0-9a-zA-Z]{1,}|::(ffff(:0{1,4}){0,1}:){0,1}((25[0-5]|(2[0-4]|1{0,1}[0-9]){0,1}[0-9])\.){3,3}(25[0-5]|(2[0-4]|1{0,1}[0-9]){0,1}[0-9])|([0-9a-fA-F]{1,4}:){1,4}:((25[0-5]|(2[0-4]|1{0,1}[0-9]){0,1}[0-9])\.){3,3}(25[0-5]|(2[0-4]|1{0,1}[0-9]){0,1}[0-9])

Table 1 presents the regular expressions (Regex) used to identify and classify Indicators of Compromise (IOCs) within the THREATMOSAIC system. These Regex patterns serve as automated rules for detecting specific types of IOCs, such as URLs, MAC addresses, and IPv6 addresses. The relevance of Table 1 lies in its role in facilitating efficient and accurate IOC detection. By leveraging these predefined Regex patterns, the system can systematically parse and extract critical cybersecurity data, enhancing threat intelligence capabilities. Including this table ensures transparency in the methodology used for IOC identification and classification.

4.4.2 Phase 2: Analysis and enrichment

This phase consists of analyzing and enriching the indicators of compromise, mainly using scanning techniques such as Virus Total or detonation such as URL Scan. For this purpose, it was divided into two sprints. The first sprint addressed the problem of storing and deleting IOCs, corresponding to Use Cases CU-0009 and CU-0010. The system was designed in such a way that once an indicator has been stored it can be removed from a project. However, if the indicator has been previously analysed, such deletion will only have an effect for the project. That is, the IOC enrichment data is not removed from the database, only its linkage to the project in question is removed. This streamlines future query and analysis operations from other projects. The second sprint of this phase corresponds to the analysis and enrichment itself, Use Case, CU-0011, taking advantage of different REST APIs available on the market. The research focused on IPv4 addresses, but later this type of analysis could be extrapolated to other types of indicators. Several libraries available in Python have been used to obtain enrichment. AbuseIPDB, IPStack, Virus Total, Ping, Whois and URLScan.io are some of the libraries used. The objective was to obtain useful information from the indicators of compromise. The result has made it possible to generate a directional graph thanks to the STIX2.1 standard and the stix2viz library used in the Frontend thanks to STIX Visualizer. In addition, the system queries the THREATMOSAIC database in search of IOCs close to the analysed indicator. It is also possible to download the indicator in STIX2.1 format or to share it on the TAXII server so that other research teams have the intelligence information. As it is modular, this type of analysis is fully scalable and has great room for improvement.

4.4.3 Phase 3: Re-injection

This phase develops the last sprint, search and re-injection that corresponds to the use cases CU-0012 and CU-0013. Taking advantage of the fact that the enriched information of the indicators of engagement is stored, a search functionality was implemented that allows showing those indicators that are stored to any type of user through queries to the MongoDB database, a programmable Google search engine was also created that facilitates the consultation of information on cyber intelligence through various specific and reliable sources related to the sector. On the other hand, taking advantage of the development of the experimental medallion server for intelligence exchange in STIX2.1 format via TAXII protocol, a functionality was created to share IOCs on this server once they have been analysed and enriched. In the future, other intelligence platforms will be able to send requests to THREATMOSAIC to obtain the intelligence stored in its database.

5 Experiments

As detailed above, IOC import, extraction, sanitisation, cleaning and classification tests were performed using a list of IOCs downloaded from the Threat Feeds repository. Specifically, for Case Study 1, we tested the IP addresses stored in the download shown in Figure 10.

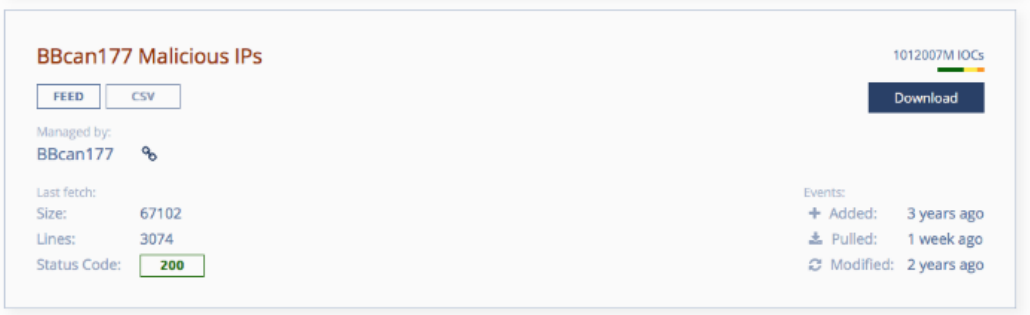

Figure 10. Threat Feeds' repository of malicious IPs. Source: ThreatFeeds [61].

This repository has 3074 lines, in which there are all kinds of information, mainly IPv4 addresses, hashes, researchers' URLs and other word strings. A Lenovo Thinkpad L480 laptop with an Intel Core i5-8250U 1.80GHz microprocessor and 8GB of RAM with a 64-bit Windows 10 Pro operating system was used to evaluate the results. The results obtained with THREATMOSAIC were compared, on the one hand, with the results obtained by two applications whose use is 100% free: IOC Extractor (cloud application) and ElevenPaths TheTHE (this application is installed OnPrem in a virtual machine with Kali Linux using the API-KEY of the Virus Total and AbuseIPDB plugins) and, on the other hand, with the results obtained by two applications in their free version but which have a paid version: Maltiverse (cloud application with different payment plans) and Anomali STAXX (this application is installed OnPrem using its virtual machine based on CentOS7, although it requires a user account in Anomali STAXX to view the enrichment of IOCs, but not to use the import functionalities). These conditions were identical for all case studies in this research. The time taken by each of the five applications to perform certain operations, mainly the processing of indicators during the clean-up and sorting phase, was evaluated. In addition, the accuracy of the results, the type and number of IOCs detected, the ability of the applications to share information and their import/export formats were considered, while the subjective part based on appearance was discarded.

5.1 Case Study 1 – Extracting IOCs from a TXT file

The file described above was used during Case Study 1. The results obtained, shown in Figure 11, show the ratio of the total number of IOCs extracted as a function of time in seconds for each application. The values on the left of the graph represent the

time in seconds and the values on the right represent the total number of IOCs extracted. In the tests performed, both IOC Extractor and ElevenPaths TheTHE require a high computational cost for many indicators.

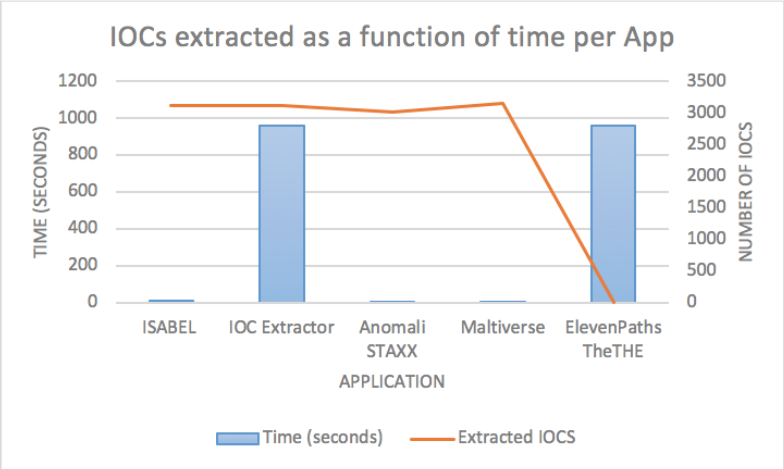

Figure 11. Total number of IOCs extracted as a function of time (in seconds) - Case Study 1.

Source: Own elaboration.

In the first case, IOC Extractor takes 957.27 seconds, that is, about 15 minutes to show the results, although it is true that, apart from extracting and classifying the IOCs, this application performs another series of processes such as evaluating whether a domain or URL is on a whitelist to ignore it as a result. It also automatically generates rules for Snort, SentinelONE and Zeek. In the case of ElevenPaths TheTHE, the operation was repeated several times, and the result was inconclusive. After more than 16 minutes, the browser displayed a crash message, so no result was obtained. As for the other two applications, Maltiverse obtained the best result, performing an immediate import in just 0.01 seconds, and Anomali STAXX came in second place, taking 6.23 seconds. THREATMOSAIC came in third place with a time of 10.49 seconds. If we analyse the types of IOCs detected, THREATMOSAIC detected a total of 3115 IOCs, coinciding with Maltiverse in its results for IPv4 and URLs with 2785 and 234 respectively, and coinciding with IOC Extractor in the number of hashes, a total of 15, in addition to those types it has detected 81 DNS domains, Figure 12.

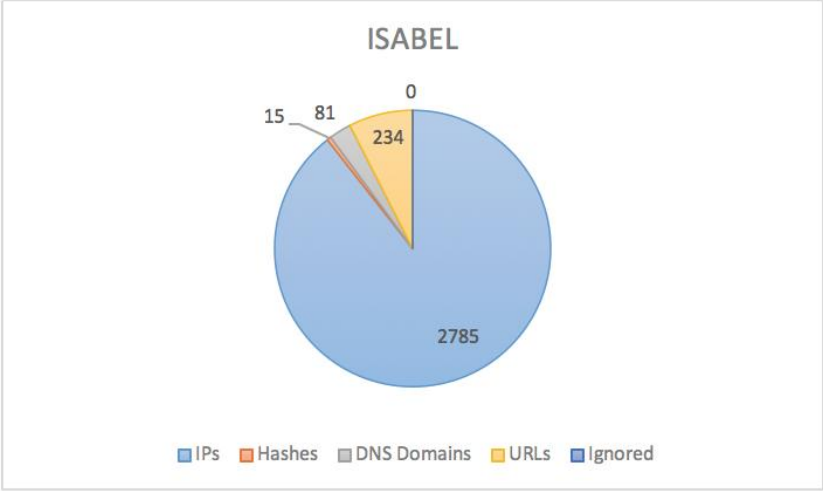

Figure 12. Number of IOCs by type detected by THREATMOSAIC - Case Study 1. Source: Own elaboration.

5.2 Case Study 2– Extraction and Enrichment of IOCs from Text

This second case study took advantage of the randomised example of IOC Extractor with a list of IPs. This time we first ran a test without enrichment and then a test with enrichment with a single IOC and compared the results for each of the applications. The 12 IOCs provided by IOC Extractor were pasted into the 5 applications and all 5 applications correctly detected and classified them as IPv4. Figure 13 shows that the best results in terms of time in seconds were obtained by ElevenPaths TheTHE and Maltiverse with 0.01 seconds, in third place was THREATMOSAIC with 0.04 seconds and in fourth and fifth place was IOC Extractor and Anomali STAXX with 1.28 seconds and 1.68 seconds, respectively.

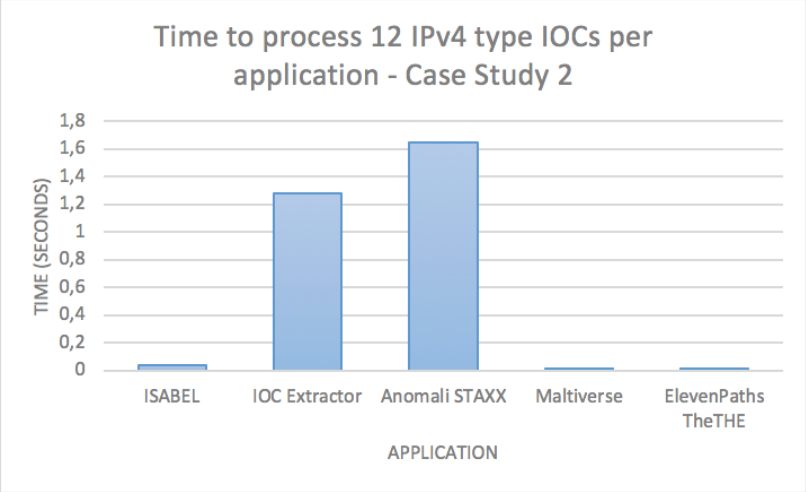

Figure 13. Time in seconds to process 12 IPv4 type IOCs for each application - Case Study 2. Source: own.

The experiment was repeated using a single IOC to evaluate the time it takes to obtain results and relevant information, i.e. enrichment that allows the CSIRTs to make decisions. The results obtained in this case were: THREATMOSAIC took on average 6.76 seconds in case the IOC is not in its database and 0.11 seconds in case it is in the database. IOC Extractor took about 0.25 seconds to display the indicator and its enrichment by extracting the information from its database. Anomali STAXX took 8.89 seconds as it queries the information in real time. Maltiverse took 0.1 seconds to display a report on the information contained in its database. And finally, ElevenPaths TheTHE took approximately 24 seconds if it is not in its database, as each plugin must be queried separately, and 0.04 seconds if the IOC is in its database.

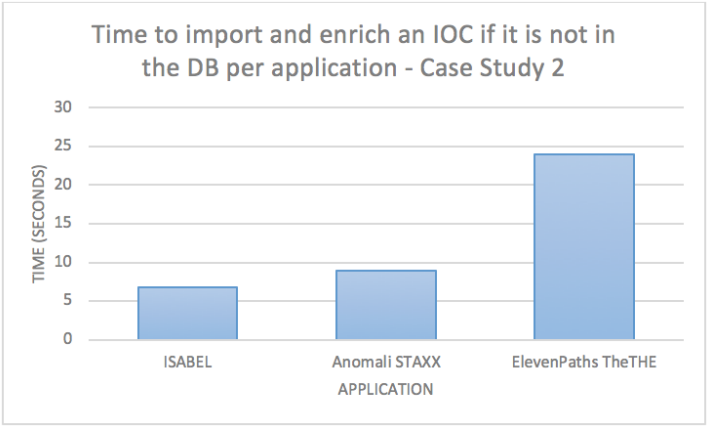

Figure 14. Time to import and enrich an IOC if it is not in the database - Case Study 2. Source: Own elaboration.

Figure 14 shows those applications that were found to perform enrichment in real time and Figure 15 shows those that do have enrichment in their database, query directly there and display it on screen without consulting other sources. From the tests carried out, it was determined that IOC Extractor and Maltiverse only query the indicators in their databases, hence their very good times when performing these types of queries. However, Anomali STAXX processes the IOC in real time and TheTHE and THREATMOSAIC, if they do not have the IOC registered in their database, query the information with the REST APIs. In the case of THREATMOSAIC it is possible to re-enrich the IOC to update that information in the database.

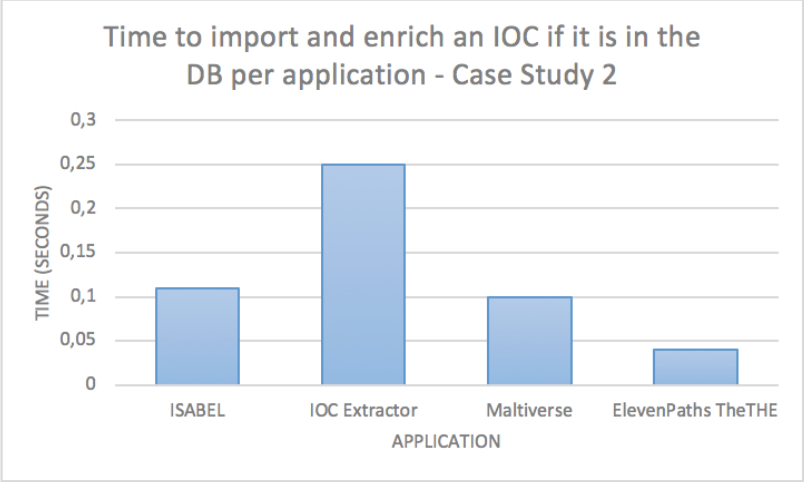

Figure 15. Time to import and enrich an IOC if it is in the database - Case Study 2. Source: Own elaboration.

5.3 Case Study 3 - Extracting IOCs from a JSON File

One of THREATMOSAIC's capabilities is that it can export the rich output in STIX2.1 format. For this case study, a threat report based on the Poison Ivy Trojan written in STIX2.1 and published on the OASIS github was used. The *.json file contains 2014 lines of code describing indicators of compromise, their relationships, their variants, the tools used to analyse them, the CVEs used, etc. The aim of this case study was to extract the indicators of compromise from this file to analyse them separately and obtain enrichment from them to counter this type of threat. The results obtained on this occasion in Figure 16 were very similar between all the applications. Of note was the fact that Anomali STAXX and Maltiverse, which are both paid applications, did not detect email-based indicators of compromise. Also, the fact that IOC Extractor ignored the URLs and DNS domains contained in its whitelists, something that an analyst can use as query information and that characterises a threat in this case.

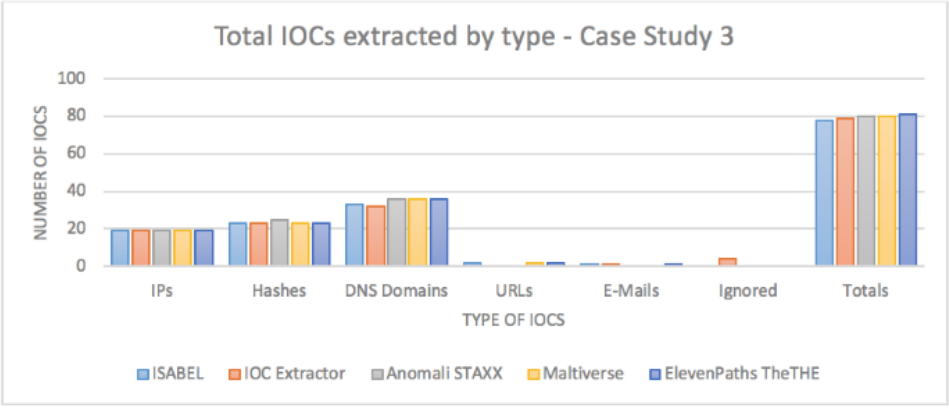

Figure 16. Total IOCs extracted by type and application - Case Study 3. Source: Own elaboration.

In terms of response time in displaying the results, THREATMOSAIC, Maltiverse and ElevenPaths TheTHE were very close, while in fourth place was Anomali STAXX with 5.62 seconds and in last place IOC Extractor with 149.81 seconds, Figure 17.

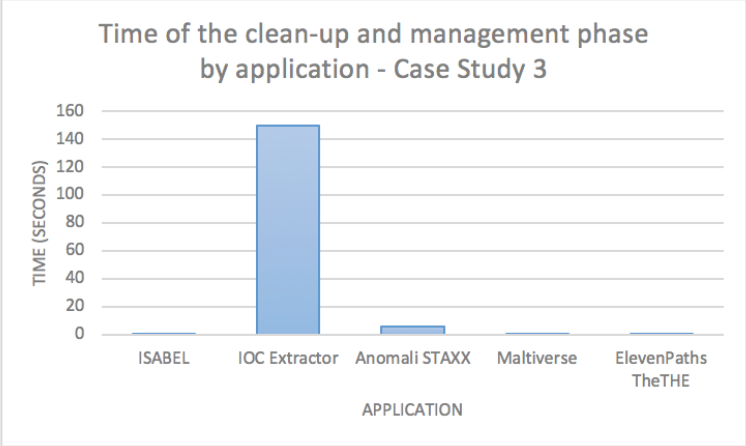

Figure 17. Time in seconds of the sanitation and management phase per application - Case Study 3.

Source: Own elaboration.

6 Discussion or Analysis of Results

Based on the case studies, THREATMOSAIC was found to be an extremely useful application for extracting indicators of compromise from a variety of sources on a massive scale. Comparatively speaking, THREATMOSAIC performed better than IOC Extractor and TheTHE in Case Study 1 and was found to be a very stable application during the clean-up and management phase. While it is true that IOC Extractor performs more tasks during this phase, its response times do not seem reasonable for a CSIRT member. Furthermore, THREATMOSAIC's response times were like those of the other two applications, Anomali STAXX and Maltiverse, both of which must pay to have all their functionalities available. During Case Study 2, THREATMOSAIC performed better than IOC Extractor and Anomali STAXX and was very close to the results obtained by Maltiverse and TheTHE when applying the clean-up and management phase to already cleaned and sorted IOCs. In the enrichment test, THREATMOSAIC obtained better results than Anomali STAXX and TheTHE when performing real-time enrichment. In the case of Anomali STAXX, the results displayed are not of great value to the analyst because an upgrade is necessary to have access to all the information. As for the case of TheTHE, it could not be assured that it was better or worse due to a temporal issue, but rather that the operations and methodology are different. However, from a practical point of view, THREATMOSAIC automates actions that in TheTHE are manual and therefore save time and resources for the analyst. However, it cannot be overlooked that overall, it has much more functionality and a higher degree of maturity as an application than THREATMOSAIC. In addition, enrichment tests were also performed when an IOC is registered in the database. THREATMOSAIC again ranked well on a par with Maltiverse and ahead of IOC Extractor. Nevertheless, it lagged behind TheTHE. In the case of Maltiverse, rather than a working environment, it has been concluded that it is a very good query tool, whose API could be used to obtain enrichment in THREATMOSAIC.

On the other hand, the enrichment generated by IOC Extractor also seems limited and cannot be considered as a framework either, although this application has features that THREATMOSAIC does not yet have but could easily incorporate. As for TheTHE and Anomali STAXX, they are two very powerful tools. However, in the case of the former, TheTHE, whether due to the researcher's lack of knowledge or other factors, shows a certain instability beyond those empirically verified. While the second, Anomali STAXX, is a limited product and shows some instability in some actions when it has been worked with. On the other hand, Case Study 3 was used to determine the effectiveness of THREATMOSAIC. All the applications obtained similar results in terms of the classification of the IOCs, except for the exceptions mentioned in that section. TheTHE was shown to be the fastest and most effective application, closely followed by both THREATMOSAIC and Maltiverse, with IOC Extractor coming in worst for the

reasons already mentioned. Finally, we compared the different options for importing/exporting IOCs and the possibility of sharing the derived intelligence through repositories or threat exchange platforms, and whether they are paid for. At this level, THREATMOSAIC contemplates the possibility of sharing IOCs in a TAXII repository, albeit on an experimental basis, which a priori is not contemplated by the other applications.

7 Conclusions

Based on the objective set out in this research, the aim was to develop software that would facilitate the work carried out by CSIRTs when processing, analyzing and sharing IOCs effectively, efficiently and at a low cost before sharing them on intelligence sharing platforms. THREATMOSAIC automates a multitude of processes linked to the analysis and management of IOCs linked to threat detection, which are carried out manually by CSIRTs daily. Firstly, from a purely functional point of view, THREATMOSAIC is a web-based application with a user-friendly interface that allows the investigator to perform analysis on suspicious IOCs in an effective and efficient manner and to record the results. The software can automatically sanitise, sort and classify IOCs of different types in large volumes and in a reasonable period from different sources and formats. In addition, it allows obtaining useful information from atomic IOCs from different sources such as Whois, AbuseIPDB, Virus Total or IPStack, generating a context that allows the analyst to make decisions with less effort, time and dedication without the need to consult all these sources one by one manually. It should also be noted that the STIX2.1 standard has been applied to the enriched IOCs, the graphic representation of which represents added value, and advantage has been taken of the deployment of a TAXII server in the experimental phase that would allow these indicators to be shared with third parties. From a technical point of view, it is multiplatform as it would be easily deployable as a cloud service in Amazon Web Services, which gives it versatility, scalability and room for improvement. Finally, it has managed to develop successful use cases that satisfactorily certify that the objectives have been met and that THREATMOSAIC is a competitive application. In short, software of these characteristics is useful and necessary in the day-to-day work of many researchers and security analysts, especially if they are members of a CSIRT. For this reason, THREATMOSAIC has great room for improvement from a functional, technical and, above all, cyber-intelligence point of view.

8 Future Work

To the validation tests, certain shortcomings have been detected at the functional level that require the implementation of corrective actions. On the other hand, there is also the possibility of adding new functionalities to the application, for example, enrichment for other types of IOCs such as hashes, MAC addresses, DNS domains or e-mail. This would increase the analysis capacity of the IOCs by adding new modules thanks to the integration of other third-party REST APIs. It would also increase the information obtained by allowing correlations to be established between the different types of indicators. Another point to consider at a functional level would be to contemplate the extraction of new types of IOCs such as CVEs or Bitcoin and Monero addresses or Defang versions of indicators such as IP addresses or URLs. Not to mention the multiple possibilities provided by the STIX2.1 format when it comes to characterising threats, a point that could be further explored and quantitatively improved. In addition to all this, the information stored in THREATMOSAIC could be combined with machine learning models, increasing the level of automation to infer from the results patterns of conduct or behaviour capable of determining TTPs of the criminal organisations that swarm the network. Finally, another type of functional improvement would be to automate the entire process from start to finish, allowing the user to introduce changes or comments that enrich the information. When looking at possible technical improvements, it should be borne in mind that there is considerable room for improvement. Visual appearance, accessibility and user experience at the presentation layer, coding and debugging of code to improve computational capacity at the logic layer, as well as the introduction of good security practices to avoid compromising dimensions such as confidentiality, integrity or availability of the software at the data layer. All these improvements would result in a more solid, robust and reliable application and above all more attractive for the analyst's work. Within the field of cyber intelligence, it must be made clear that although it is often based on IOCs, IOCs alone are not cyber intelligence. The job of analysts is to foresee, anticipate, detect and mitigate security breaches in time based on highly detailed intelligence reports where IOCs may play a role, but they are not the whole, they are a part of it. Joining forces and sharing this type of intelligence between the different CSIRTs is vital and this application has great room for improvement to facilitate this process at that level. Therefore, extending the capabilities and functionalities of THREATMOSAIC would allow for the analysis of patterns of behaviour related to criminal behaviour in the future, making the

work carried out by investigators in the sector much faster and more effective. Finally, it is vital to raise awareness of computer security in our society to hinder the work of all those criminal groups that have made fraud, extortion and blackmail their livelihood thanks to the apparent anonymity guaranteed by the Internet. We know that it is possible to implement expensive security systems, fast and powerful applications to process IOCs or protect infrastructures, or to invest millions of euros in intelligence, but we must be aware that the weakest link in this whole chain, which is the security management process, will always fall on the same indicator of compromise, on the same asset: the user. That user who, at the end of the day, hides behind him an imperfect person with his virtues and defects, so typical of human nature.

Acknowledgments

This work has been funded by the call for grants for research stays abroad 2024/2025 of the International University of La Rioja (UNIR). Grant PID2023-150566OB-I00 (AUDITIA-X) funded by MCIN/AEI /10.13039/501100011033 and by ERDF/EU.

References

[1] S. Gatlan, Ryuk ransomware hits 700 Spanish government labor agency offices, 10 03 2021. Available: https://www.bleepingcomputer.com/news/security/ryuk-ransomware-hits-700-spanish-government-labor-agency-offices/.

[2] Z. M. Smith, E. Lostri y J. A. Lewis, The Hidden Costs of Cybercrime, McAfee, San José, 2020.

[3] O. Sviatun, O. Goncharuk, C. Roman, O. Kuzmenko y I. Kozych, Combating Cybercrime: Economic and Legal Aspects, *Wseas transactions on business and economics,* vol. 18, pp. 751-762, 21 04 2021.

[4] M. Rego Fernández y P. P. Pérez García, El intercambio de información de ciberamenazas, de *Ciberseguridad: la cooperación público-privada*, I. E. d. E. Estratégicos, Ed., Madrid, Ministerio de Defensa - Secretaría General Técnica, 2016, pp. 139-169.

[5] D. Pérez, IOCs, una palabra de moda, un tema caliente. Pero, ¿realmente conocemos sus capacidades?, 25 03 2016. Available: https://www.pandasecurity.com/es/mediacenter/seguridad/iocs-y-sus-capacidades/.

[6] M. Bromiley, SANS 2019 Incident Response (IR) Survey: It's Time for a Change, *SANS Institute - Information Security Reading Room,* pp. 1-16, 2019.

[7] Virus Total, Virus Total, 20 06 2021. Available: https://www.virustotal.com/.

[8] urlscan.io, About - urlscan.io, 15 6 2021. Available: https://urlscan.io/about/.

[9] OASIS, Introduction to STIX, 01 04 2021. Available: https://oasis-open.github.io/cti-documentation/stix/intro.

[10] OASIS, Introduction to TAXII, 01 04 2021. Available: https://oasis-open.github.io/cti-documentation/taxii/intro.

[11] M. Khanam, M. Lutf, A. Kumar, A. Ahmed y T. Ara, Threat Intelligence Sharing: A Survey, *JSAC: Journal of Applied Science and Computations,* pp. 1811-1815, 2018.

[12] P. Skalický y T. Palasiewicz, Intelligence Preparation of the Battlefield as a Part of Knowledge Development, *International conference KNOWLEDGE-BASED ORGANIZATION.,* p. 23, 01 2017.

[13] W. Young, A Deep Dive into the Firepower Manager, de *Cisco Live!*, Cancún, 2017.

[14] R. Stillions, The DML model, 22 04 2014. Available: http://ryanstillions.blogspot.com/2014/04/the-dml-model_21.html.

[15] Cyber Threat Alliance (CTA), Cyber Threat Alliance, 01 04 2021. Available: https://www.cyberthreatalliance.org/.

[16] A. Zibak y A. Simpson, Cyber Threat Information Sharing: Perceived Benefits and Barriers, de *ARES '19: Proceedings of the 14th International Conference on Availability, Reliability and Security.*, Canterbury, 2019.

[17] IBM, IBM X-Force Exchange, 01 03 2021. Available: https://www.ibm.com/products/xforce-exchange.

[18] Anomali, ThreatStream, 05 03 2021. Available: https://www.anomali.com/products/threatstream.

[19] SolarWinds, Security Event Manager, 04 03 2021. Available: https://www.solarwinds.com/security-event-manager.

[20] Palo Alto Networks, AutoFocus - World's highest-fidelity Contextual Threat Intelligence - Palo Alto Networks, 24 4 2021. Available: https://www.paloaltonetworks.com/cortex/autofocus.

[21] LogRhythm, Watch the LogRhythm Threat Lifecycle Management Demo, 04 04 2021. Available: https://logrhythm.com/rapidly-detect-and-respond-to-cyber-threats-with-threat-lifecycle-management-demo/.

[22] FireEye, Mandiant Advantage Platform, 01 04 2021. Available: https://www.fireeye.com/mandiant/advantage.html.

[23] LookingGlass, LookingGlass Cyber Solutions Inc., 01 04 2021. Available: https://www.lookingglasscyber.com.

[24] AT&T Cybersecurity, Unified Security Management (USM), 01 04 2021. Available: https://cybersecurity.att.com/products/usm-anywhere.

[25] MISP, MISP - Open Source Threat Intelligence Platform & Open Standards For Threat Information Sharing (formely known as Malware Information Sharing Platform), 10 03 2021. Available: https://www.misp-projects.org.

[26] O. Catakoglu, M. Balduzzi y D. Balzarotti, Automatic Extraction of Indicators of Compromise for Web Applications, de *Proceedings of the 25th International Conference on World Wide Web*, Republic and Canton of Geneva, CHE, 2016.

[27] A. Villalón, Los IOC han muerto, ¡larga vida a los IOC!, 28 07 2020. Available: https://www.securityartwork.es/2020/07/28/los-ioc-han-muerto-larga-vida-a-los-ioc/.

[28] D. Bianco, The Pyramid of Pain, 01 03 2013. Available: http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html.

[29] J. Maynard, The Canadian Bacon: Cisco Security and the Pyramid of Pain, 26 08 2020. Available: https://gblogs.cisco.com/ca/2020/08/26/the-canadian-bacon-cisco-security-and-the-pyramid-of-pain/.

[30] R. Danyliw, J. Meijer y Y. Demchenko, IODEF, 01 04 2021. Available: https://tools.ietf.org/html/rfc5070.

[31] Mandiant, Contribute to mandiant/OpenIOC_1.1 development by creating an account on GitHub, 02 04 2021. Available: https://github.com/mandiant/OpenIOC_1.1.

[32] OASIS, OASIS Cyber Threat Intelligence (CTI) TC | OASIS, 01 02 2021. Available: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=cti.

[33] MAEC, MAEC 5.0 | MAEC Project Documentation, 15 02 2021. Available: http://maecproject.github.io/releases/5.0/.

[34] IOC Bucket, IOC Bucket, 01 04 2021. Available: https://www.iocbucket.com.

[35] Pulsedive LLC, Free threat intelligence feeds - threatfeeds.io, 01 04 2021. Available: https://threatfeeds.io.

[36] Anomali, STAXX | Herramientas STIX / TAXII gratis, 02 12 2020. Available: https://www.anomali.com/es/resources/staxx.

[37] Carnegie Mellon University, "Introduction to Trheat Hunting," Carnegie Mellon University, p. 127, 2023. Available: https://apps.dtic.mil/sti/trecms/pdf/AD1214459.pdf.

[38] A. Gomez, H. Sanchez, D. Gil y J. Lopez, Maltiverse, 10 05 2021. Available: https://www.maltiverse.com.

[39] The Hive Project, The Hive Project, 29 06 2021. Available: https://thehive-project.org.

[40] IOC Extractor, IOC Extractor, 01 01 2021. Available: https://iocextractor.com.

[41] Open CTI, Open CTI, 01 06 2020. Available: https://opencti.io.

[42] A. Balci, D. Ungureanu y J. Vondruska, Malware Reverse Engineering Handbook, CCDCOE - NATO Cooperative Cyber Defence Centre Of Excellence, Tallinn, 2020.

[43] F. Weimer, Passive DNS Replication, 30 04 2004. Available: https://static.enyo.de/fw/volatile/pdr-draft-11.pdf.

[44] L. Bilge, E. Kirda, C. Kruegel y M. Balduzzi, EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis, *Proceedings of the Network and Distributed System Security Symposium, NDSS 2011,* pp. 1-17, 2011.

[45] M. Kuyuma, Y. Kakizaki y S. Ryoichi, Method for Detecting a Malicious Domain by using WHOIS and DNS features, de *The Third International Conference on Digital Security and Forensics (DigitalSec2016)*, Malasia, 2016.

[46] B. G. Kang, J. S. Yoon, L. Min Wook y L. Sang Jin, Automatic Creation of Forensic Indicators with Cuckoo Sandbox and Its Application, *Journal of the Korea Information Processing Society: Computers and Communication Systems,* vol. 5, nº 11, pp. 419-426, 30 11 2016.

[47] L. Rudman y B. Irwin, Dridex: Analysis of the traffic and automatic generation of IOCs, de *2016 Information Security for South Africa (ISSA)*, Johannesburg, 2016.

[48] A. Rodríguez García, IOCSeeder. Análisis dinámico de malware mediante sandboxing para generación de IOC's - Think Big Empresas, 04 02 2018. Available: https://empresas.blogthinkbig.com/iocseeder-analisis-dinamico-de-malware/.

[49] ANY.RUN, ANY.RUN, 30 07 2021. Available: https://any.run.

[50] E. Domínguez de la Iglesia, ¿Qué es el reversing de malware?, 23 04 2020. Available: https://www.campusciberseguridad.com/blog/item/140-que-es-el-reversing.

[51] G. Palazolo y F. Duarte, Reverse Engineering Dridex and Automating IOC Extraction, 18 09 2020. Available: https://www.appgate.com/blog/reverse-engineering-dridex-and-automating-ioc-extraction.

[52] P. Harrington, Machine Learning in Action, USA: Manning Publications Co., 2012, p. 384.

[53] Stack Overflow, Stack Overflow Survey 2020, Stack Overflow, World Wide Web, 2020.

[54] J. Richardson, R. Sallam, K. Schlegel, A. Kronz y J. Sun, Magic Quadrant for Analytics and Business Intelligence Platforms, 11 02 2020. Available: https://www.gartner.com/en/documents/3980852/magic-quadrant-for-analytics-and-business-intelligence-p.

[55] C. Kotas, T. Naughton y N. Imam, A comparison of Amazon Web Services and Microsoft Azure cloud platforms for high performance computing, de *2018 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, 2018.

[56] N. Mbeiri, Caracterización del Ransomware Ryuk utilizando las herramientas de CTI, de *IntelCon 2021*, Madrid, 2021.

[57] IEEE, IEEE Recommended Practice for Software Requirements Specifications, *IEEE Std 830-1998,* pp. 1-40, 20 10 1998.

[58] R. Wieruch, A Firebase in React Tutorial for Beginners, 20 11 2018. Available: https://www.robinwieruch.de/complete-firebase-authentication-react-tutorial.

[59] M. Schwartz, https://github.com/mosesschwartz/extract_iocs, 10 12 2017. Available: https://github.com/mosesschwartz/extract_iocs.