Project Deliverable

D4.1
Techniques and tools for OSINT-based threat analysis

Project Number	700692
Project Title	DiSIEM – Diversity-enhancements for SIEMs
Programme	H2020-DS-04-2015

Deliverable type	Report
Dissemination level	PU
Submission date	August 31st, 2017
Resubmission date	May 31st, 2018

Responsible partner	FCiências.ID
Editor	Pedro M. Ferreira
Revision	2.0

The DiSIEM project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 700692.

Editor
Pedro Ferreira, FCiências.ID

Contributors
Pedro Ferreira, FCiências.ID
Alysson Bessani, FCiências.ID
Fernando Alves, FCiências.ID
Eunice Branco, FCiências.ID
Ana Respício, FCiências.ID
João Alves, FCiências.ID
Susana Gonzalez, Atos
Mario Faiella, Atos
Gustavo Gonzalez, Atos
Abdullahi Adamu, DigitalMR
Ilir Gashi, City

Version History

Version	Author	Description
1.0	FCiências.ID, Atos, DigitalMR, City	First version of the document, submitted to the EC
2.0	FCiências.ID	Corrections of the minor problems requested by the reviewers in the first period review. In particular, we removed the list of tweeter cybersecurity-related accounts followed by project partners to avoid possible privacy conflicts. Submitted to EC.

Executive Summary

This report presents an in-depth analysis of security-related OSINT data sources and how the information from these sources can be extracted, including a description of tools and methods that can be employed for this. Relevant open-source and paid tools, as well as services available, are identified and described.

A complete list of OSINT sources selected by the DiSIEM industrial partners, currently being collected in the project, is also provided. Its main purpose is to create realistic case-studies that enable a sound evaluation of the technologies being developed. The deliverable also describes initial work on the models and techniques that can be used to process OSINT data for predicting threats against a given organisation's IT infrastructure.

Additionally, the techniques that can be used to express and share gathered OSINT in a standardized way are also reviewed. Finally, the deliverable ends by proposing an architecture for infrastructure-aware OSINT integration, i.e., how to integrate relevant OSINT with events from the infrastructure and provide a related threat score.

Overall, the main results of this deliverable are:

- A taxonomy of the types of OSINT sources considered on the project;
- The various sources in use by the partners of the project;
- A list of relevant tools and services that can be used for the processing and analysis of OSINT
- A description of the state of the art using security-related OSINT;
- The OSINT processing tools being developed on the project, and their objectives;
- How to integrate OSINT events in the SIEMs.

Table of Contents

List of Figures

List of Tables

1 Introduction

Cybersecurity is a matter of growing concern as cyber-attacks cause loss of income, sensitive information leaks, and even vital infrastructures to fail. To properly protect an infrastructure, a security analyst must have timely information about security threats to the IT infrastructure and the latest news in terms of updates, patches, mitigation measures, vulnerabilities, attacks, and exploits.

There are two major ways of obtaining security news feeds. One is to purchase a curated feed from a specialized company such as SenseCy[1] or SurfWatch.[2] Another is to collect Open Source Intelligence (OSINT) from various sources available on the Internet. In summary, OSINT is information publicly available on the news and on the web. Examples of cybersecurity-related OSINT feeds are Cisco Security Advisory[3] and Threatpost.[4]

Collecting and processing OSINT is becoming a fundamental approach for obtaining cybersecurity threat awareness. Recently, the research community has demonstrated that many different types of useful information and Indicators of Compromise (IoC) can be obtained from OSINT [LIA16, SAB15, ZHU16]. Besides these research oriented efforts, all Security Operation Centres (SOC) analysts try to be updated about possible threats against the IT infrastructure of their organizations by following cybersecurity OSINT. Nevertheless, skimming through various news feeds is a time-consuming task for any security analyst. Furthermore, an analyst is not guaranteed to find news relevant to the IT infrastructure he/she oversees. Therefore, tools are required not only to collect OSINT, but also to process it to filter only the relevant parts for the SOC analysts, thus decreasing the amount of information and consequently the time required to analyse it and act upon it. When appropriate, the filtered information must be further processed to extract IoCs.

This report provides a taxonomy of OSINT sources readily available and gives a comprehensive list of OSINT data sources that are currently being considered in the scope of DiSIEM. A review on existing techniques and tools available for OSINT processing is given in the document, as well as preliminary results on the infrastructure-related OSINT processing approaches being followed in DiSIEM.

The ability to collect and process OSINT is often not enough. Threat intelligence must be expressed and then shared using specific standards, allowing involved parties to speed up processing and analysis phases of received information, achieving interoperability among them. Additionally, the gathered OSINT should be integrated with events originating within the organisation's IT infrastructure and given a threat score indicating its severity. This document also discusses a

[1] https://www.sensecy.com/
[2] https://www.surfwatchlabs.com/threat-intelligence-products/threat-analyst
[3] https://tools.cisco.com/security/center/psirtrss20/CiscoSecurityAdvisory.xml
[4] https://threatpost.com/feed/

standard designed to transmit OSINT data that will be used to create data flows among the software components designed in the project, and to the SIEMs.

The final contributions presented in this document are a proposed system architecture and threat score for context-aware OSINT integration.

1.1 Organization of the Document

Chapter 2 is devoted to presenting the various types of OSINT that are available on the Internet as well as common extraction and storage tools that can be used. The section ends by presenting a list of OSINT sources that are being collected for the development of DiSIEM tools. Chapter 3 presents related work on techniques and tools for OSINT analysis and existing tools for that purpose. Preliminary results on OSINT processing approaches that are being followed in DiSIEM are given in Chapter 4. Then, integration of security-related OSINT with security events from the organisation IT infrastructure is approached in Chapter 5. Finally, Chapter 6 presents a summary of the work and draws some conclusions.

2 OSINT Data Sources

2.1 Types of sources

In this section we describe the data sources used when gathering security-related OSINT. Table 1 presents a taxonomy classifying sources as structured and unstructured, keeping a separate class for the dark web even though it is considered an unstructured source. The table presents examples of each source type, as well as the technologies required to collect data from those sources. The three major classes are:

Structured data sources: Resources that provide structured data, in a well-defined format. The data obtained from these sources comes in a machine parsable format.

Unstructured data sources: Feeds that provide unstructured data where the main content is in free text format. Although this data type requires further processing, feeds in text format (such as news posts) are typically more information rich.

Dark web: The "dark side of the Internet", a place known for hacker sites and forums, and exploit marketplaces. Both are rich information sources for malicious activity, mostly unstructured.

	Structured data sources	Unstructured data sources	Dark web
Examples	- IP whitelists/ blacklists - CVE	- News sites - Twitter	- Forums - Marketplaces
Required technologies	- Feed/web scraper - Parser	- Feed/web scraper - Natural Language Processing (NLP) tools - Machine learning techniques	- Dark web access - Dark web scraper - NLP tools - Machine learning techniques

Table 1 - Taxonomy of OSINT sources.

2.1.1 Structured data sources

Organized data sources present information that is machine-parsable, and thus can be directly fed to a machine. These sources usually provide an API for programmatic access to their content.

Vulnerability/exploit sources. Organized sources describing vulnerabilities and/or exploits related only to threats that have been confirmed. In case of vulnerability databases, each vulnerability is described using several numeric fields (such as severity), and a text description.

Organized datasets provide the most reliable information since their content has been officially confirmed. This reliability comes at a price; usually, there is a time lapse between the detection of a vulnerability and its presence in this type of database.

Two of the most important structured vulnerability databases are the National Vulnerability Database (NVD)[5] and Common Vulnerabilities and Exposures (CVE).[6] Others include the Exploit Database[7] and Vulners.[8] The NVD belongs to the U.S. government and describes checklists, security related software flaws, misconfigurations, product names, and impact metrics. The CVE provides a structured database for publicly known information-security vulnerabilities and exposures. The vulnerabilities stored there are described in various components, as well as references to the vulnerabilities.

IP and rules sources. In the case of blacklists or whitelists, or sets of rules (e.g., firewall rules) the data is available in text files with an IP/rule per line. Each line can be fed directly to the corresponding software. There are many sources of IP lists and rules, such as the ones presented on Appendix A.

2.1.2 Unstructured data sources

Unstructured sources provide text data describing events of all sorts, including security ones. Blogs and news may contain more information (e.g., a quick fix to a vulnerability), but pose a hard challenge for automated processing since extracting concepts from free text is still a Natural Language Processing (NLP) challenge. Therefore, as appealing as they may be, using them as OSINT sources is far from trivial. Nevertheless, some authors show it is possible to collect data from technical blog posts and scientific literature, since technical writing tends to have a stable structure and much less ambiguity when comparing to other types of writing [LIA16, ZHU16].

One of the unstructured data sources used in DiSIEM is Twitter,[9] a micro-blog service where users can publish text and media content. Tweets tend to provide concise information due to the 140-character limit. Therefore, tweets are attractive for publishing quick status updates; news sites, bloggers, and other feeds post tweets containing the post's title to increase the visibility of the content they produce. Twitter is a popular feed since a quick review of tweet titles provides an overview of current news and trends. Tweets are also attractive for automated processing, as small concise messages are simpler to process than large texts.

[5] https://nvd.nist.gov/

[6] https://cve.mitre.org/

[7] https://www.exploit-db.com/

[8] https://vulners.com/#help

[9] https://twitter.com/

2.1.3 Dark web

Accessible only using anonymity tools (e.g., TOR network[10]), the dark web offers anonymity to the users accessing it and to the services hosted on it. Therefore, it is the ideal place for buying, selling, and discussing all types of illegal commodities and services. This is also true for botnets, exploits, viruses and all kinds of malicious IT services.

The dark web is a known place where exploit discussion and development happens. Collecting information about threats during their development phase or about threats for sale which have not been used yet is extremely valuable, as it allows defenders to act before the attackers. In fact, this approach has been successfully undertaken by Nunes et al. [NUN16], who obtained data on zero day vulnerabilities on dark web marketplaces and hacker forums.

2.2 OSINT extraction tools

In terms of collecting information freely from the various social media sources, the state-of-the-art uses crawlers in conjunction with parsers to extract information from the web pages of blogs, forums, marketplaces, and other relevant sites [NUN16, KER15, JEN16]. Some sources of data such as specific security websites or blogs will require gathering data using a custom-built crawler in conjunction with a parser.

For other data sources such as Twitter, Instagram, or news feeds, companies like DigitalMR (a member of the DiSIEM consortium), who have experience collecting OSINT from structured and unstructured sources, can provide historical data.

Real-time access and historical data from the complete feed of most social networks is commercially available from providers such as Gnip[11] or DataSift.[12] The free alternative consists in using APIs provided by the social media networks to access information, although usually there is no access to the full stream of data.

DigitalMR's Listening247 platform is a social media monitoring and analytics platform that is being used in DiSIEM to collect and filter relevant security-related OSINT. Listening247 provides access to social media networks data, blogs, forums or web-sites, both historic and present. This data can be accessed with carefully formed queries with specific keywords which populates DigitalMR's Elasticsearch[13] database with relevant data from all the various sources. Considering DiSIEM, there will be a need for specialized queries with specific keywords for the project, which will yield relevant data to the infrastructure that partners are interested in protecting.

[10] https://www.torproject.org/
[11] https://gnip.com/
[12] http://datasift.com/
[13] https://www.elastic.co/products/elasticsearch

Forming these queries requires careful attention and understanding of the domain, something that has been perfected at DigitalMR. Just using a query with a keyword for 'Windows' (i.e., the operating system) will yield data about 'windows' (i.e., for buildings), and other more abstract uses of the word.

Scraped data using the custom-built crawlers can be done at intervals that overlap with the intervals of the data from other Listening247 OSINT sources to allow for aggregating the data from the two data gathering pipelines. Elasticsearch can be used as a storage for the data from both pipelines.

There are pre-processing and noise filtering steps that will be carried out on the raw data as well on the Listening247 platform's custom pipeline for this project. In the pre-processing step, in compliance with the ethics advice from the advisory board, an additional step will involve anonymizing the data to strip away any information that might violate the privacy of the users of these platforms. In the noise filtering step, a noise model will be created based on annotated training data to filter out irrelevant data. A related work by Nunes et al. [NUN16] also included a noise filtering step which used a classifier for filtering out irrelevant data. There is a huge amount of data on the internet and reducing the data to only the relevant, saves both time and cost needed to store and process irrelevant data.

Data will be aggregated by timestamp, so that data within the same time interval ends up in the same time slice. These slices of data can be built by employing cloud services such as AWS ElasticMapReduce,[14] or local processing frameworks such as Apache Spark[15] which will aggregate the various forms of data for the next stage of processing. This not only gives the data a context of what is happening in the various sources of OSINT data sharing the same time slice, it also gives a context of how all the content is changing over time. Other approaches that aggregate over some relevant property of data could also be explored before implementation, if necessary. For example, lagging some of the data sources that generate data faster (e.g., Twitter) so that the same information is not split up in different time slices.

2.3 Data sources considered in DiSIEM

Besides security-related machine-parsable information such as IP black or white lists or firewall rules, and tweets from specific security-related twitter accounts, DigitalMR's Listening247 platform will be used in DiSIEM to gather information from social media sources and unstructured sources such as blogs, forums or web-sites. Listening247 has been used successfully for market research and can be tailored for cyber-security purposes. Regarding the information contained in databases like NVD or CVE, DiSIEM uses the vepRisk tool [AND17],[16] which extracts, parses and stores data from these and other public repositories.

[14] http://docs.aws.amazon.com/ElasticMapReduce/latest/API/Welcome.html
[15] https://spark.apache.org/
[16] http://veprisk.city.ac.uk/main/

Generically, DigitalMR can collect data of various types including text and images from a variety of sources which range from blogs, social networks, news, darknet, boards/fora and other openly available data on the Internet for market research (see Figure 1). The data is unstructured, and has different velocities depending on the source. An article from the dailymail[17] highlighted some statistics from Internet livestats.com [LIB16], showing that there are about 7,620 tweets per second, 790 photos uploaded to Instagram per second, and 1,259 posts to Tumblr per second. For other sources like boards, news, and blogs, which are usually longer in length, this sort of velocity is unlikely.

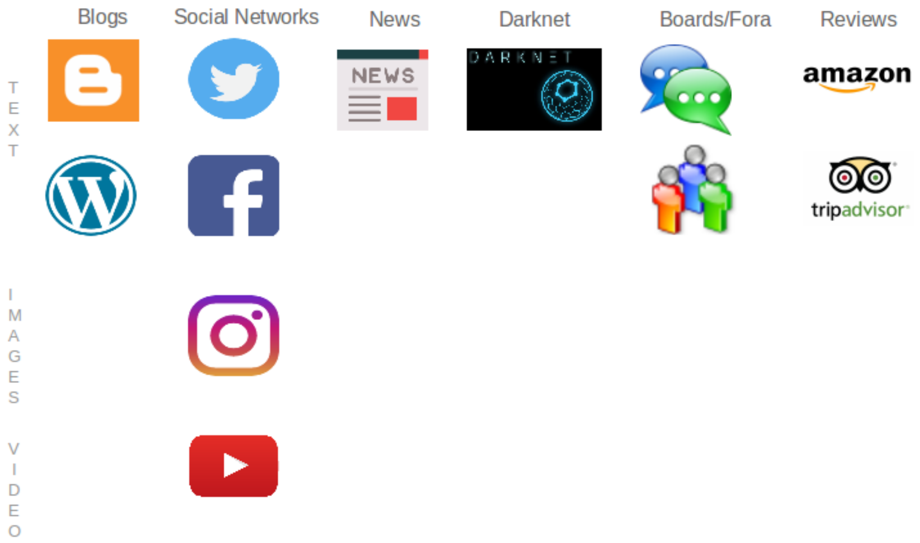

Figure 1 - DigitalMR data sources.

These are typically tagged with information pertaining to relevance, sentiment, and emotion, for market research. It may also be tagged with information that classifies these data in terms of a taxonomy which can be used to organize the data in a hierarchical format by their topic. For example, a tweet about two people drinking Pepsi while watching a football match will be classified to be an 'occasion', and specifically a 'sport' occasion. Essentially, this is a form of hierarchical clustering which can also be done for cyber-threats.

Interestingly, there is already a taxonomy of the types of cyber threats developed by ENISA (European Agency for Network and Information Security), including asset exposure and vulnerability exploitation [CEB10, MAR16]. Such information could also be tagged with the training data meant for the cyber threat prediction model which will make the reports of cyber threats more specific. The tags will also allow the cyber threat predictor to learn and predict future events with more specificity. Furthermore, another advantage of tagging in relation to a taxonomy is that it can also be used to identify specific trends, e.g. seasons when some threats are more prominent than others, and any other related trends.

[17] http://dailym.ai/28YNsq9

For information related to vulnerabilities, exploits, and patches, the vepRisk tool is used in DiSIEM. The tool has backend modules that mine, extract, parse and store data from public repositories of vulnerabilities, exploits and patches. vepRisk serves as a knowledge base for public security data and provides a web interface for analysing and visualizing the underlying data. It provides functionality for analysing relationships between the different security risk factors in public security data. Currently six different vulnerability data sources are considered: NVD, Security database,[18] CVE, CVE Details,[19] Security focus,[20] and CXSECURITY.[21] Additionally, vepRisk collects data from various vendor patch sources (e.g., Microsoft, Debian, SUSE Linux, Cisco) and exploits from Exploit database.[22]

The industrial partners of DiSIEM that operate SIEM platforms provided a list of OSINT sources that their security analysts regularly monitor to receive events relevant to their protected infrastructure. These sources are being continuously collected to form sufficiently large representative data sets that enable researching efficient OSINT processing and analysis technologies. During the project execution, the list of sources may be updated according to the requirements of the tools being developed. A comprehensive list with all OSINT sources being collected is presented in Appendix A.

Source	Example
Twitter	https://twitter.com/threatmeter/status/887390382094516229 @threatmeter: “Vuln: RETIRED: Linux Kernel 'saa7164-bus.c' Local Privilege Escalation Vulnerability http://ift.tt/2tcvTsM”
News sites	DarkReading: http://www.darkreading.com/cloud/zero-day-exploit-surfaces-that-may-affect-millions-of-iot-users/d/d-id/1329380? “A zero-day vulnerability dubbed Devil's Ivy is discovered in a widely used third-party toolkit called gSOAP. Millions of IoT devices relying on widely used third-party toolkit gSOAP could face a zero-day attack, security firm Senrio disclosed Tuesday, which dubbed the vulnerability Devil's Ivy. <...>”
Expert blogs	Schneier on Security: https://www.schneier.com/blog/archives/2017/07/forged_document_1.html

[18] https://www.security-database.com/
[19] http://www.cvedetails.com/
[20] http://www.securityfocus.com/
[21] https://cxsecurity.com/
[22] https://www.exploit-db.com/

	“Forged Documents and Microsoft Fonts A set of documents in Pakistan were detected as forgeries because their fonts were not in circulation at the time the documents were dated.”
Security vendor blogs	FireEye Threat Research Blog https://www.fireeye.com/blog/threat-research/2017/05/cyber-espionage-apt32.html “Cyber Espionage is Alive and Well: APT32 and the Threat to Global Corporations <...>”
IPs for whitelists	Bambenek consulting http://osint.bambenekconsulting.com/feeds/c2-ipmasterlist.txt “5.101.153.16,IP used by banjori C&C,2017-07-1818:05, http://osint.bambenekconsulting.com/manual/banjori.txt 23.104.241.95,IP used by banjori C&C,2017-07-18 18:05, http://osint.bambenekconsulting.com/manual/banjori.txt <...>”
IPs for blacklists	abuse.ch ZeuS Tracker: https://zeustracker.abuse.ch/blocklist.php?download=domainblocklist “039b1ee.netsolhost.com 03a6b7a.netsolhost.com 03a6f57.netsolhost.com <...>”
Domains/ botnets	abuse.ch Ransomware Tracker http://ransomwaretracker.abuse.ch/downloads/RW_DOMBL.txt “25z5g623wpqpdwis.onion.to 27c73bq66y4xqoh7.dorfact.at <...>”
Snort/ Suricata rules	Emerging threats http://rules.emergingthreats.net/blockrules/emerging-botcc.portgrouped.suricata.rules “alert tcp $HOME_NET any -> 50.116.1.225 22 (msg:"ET CNC Shadowserver Reported CnC Server Port 22 Group 1"; flow:to_server,established; flags:S; reference:url,doc.emergingthreats.net/bin/view/Main/BotCC; reference:url,www.shadowserver.org; threshold: type limit, track by_src, seconds 360, count 1; classtype:trojan-activity; flowbits:set,ET.Evil; flowbits:set,ET.BotccIP; sid:2405000; rev:4687;) <...>”

Bro	No example available

Firewall rules	Emerging threats
	http://rules.emergingthreats.net/fwrules/emerging-IPTABLES-DROP.rules
	`“$IPTABLES -N ETBLOCKLIST`
	`$IPTABLES -I FORWARD 1 -j ETBLOCKLIST`
	`$IPTABLES -I INPUT 1 -j ETBLOCKLIST`
	`<...>”`

Malware	Virus total
	https://www.virustotal.com/en/file/3232fb8c336d280b8552a0f796a3b7e6ef2a67b603d9716a7053b17596e8b24c/analysis/
	`“SHA256:`
	`3232fb8c336d280b8552a0f796a3b7e6ef2a67b603d9716a7053b17596e8b24c`
	`File name:  microsoft.visualbasic.dll`
	`Detection ratio: 0 / 64`
	`Analysis date: 2017-07-28 17:49:37 UTC ( 53 minutes ago )`
	`<...>”`

IP reputation	The CINS Score
	http://cinsscore.com/list/ci-badguys.txt
	`“1.1.198.38`
	`1.9.13.156`
	`1.9.135.197`
	`<...>”`

Yara rules	Yara-Rules
	https://github.com/Yara-Rules/rules/blob/master/CVE_Rules/CVE-2010-0887.yar
	`“rule JavaDeploymentToolkit`
	`{`
	`   meta:`
	`   ref = "CVE-2010-0887"`
	`(…)”`

Table 2 - The various OSINT sources used in the project and an example of each.

3 Techniques and tools for OSINT analysis

3.1 Related work

In this chapter we present an overview of research work that uses OSINT in a security context, divided into seven sections accordingly to the main objectives of these works.

3.1.1 Collecting infrastructure-specific OSINT

Most work described in this section uses Twitter as the OSINT data source, and follows the same general principle. First, they obtain from the user a keyword set which is then used to select tweets containing one or more keywords. This approach sets a primary filter for gathering only possibly relevant content for the user. Then, another technique is used to classify the tweets as relevant or not: Ritter et al. [RIT15] compare a few machine learning techniques, with Expectation-Maximization (EM) [MOO96] obtaining the best results; Mittal et al. [MIT16] use Naive Bayes [ZAK14], while Correia et al. [COR16] use Support Vector Machines (SVM) [ZAK14], and Santos et al. [SAN13] use plain-text searches (Apache Lucene[23]) for a cluster-like approach.

Each of these approaches present unique elements. Santos and co-workers filter tweets according to the following criteria: written in English, correctly formed, and containing URLs of websites focused on security news. Then, the tweets are clustered using a specific similarity measure that considers the tweet size and the number of equal words. To avoid presenting spam messages as relevant, a cluster is considered relevant only if the tweets contained were posted by a significant number (10) of different users. Their results are evaluated in two aspects: 1) is it possible to remove spam messages from the legitimate content? 2) is it possible to select the security tweets with most relevance? The applied techniques reduced the amount of spam messages by 22% on average, and of the messages presented as security relevant, 61.3% were selected correctly.

Instead of collecting tweets by keyword, Correia et al. gathered tweets only from security accounts to reduce the amount of non-security related tweets. The collected tweets were filtered by the keyword set, and then manually labelled; about 10 thousand tweets were manually classified. In this work two feature extraction methods were compared: TF-IDF (Term Frequency – Inverse Document Frequency) [ZAK14] and word2vec [W2V]. The tweets were classified using an SVM classifier. Correia et al. achieved high true positive rates (around 90%) with low false positive and false negative rates.

The base of Mittal et al.'s [MIT16] work is a knowledge base, created using security concepts. Further, they use external ontologies for word disambiguation (e.g., apple refers to the fruit or the company). Key concepts from the tweets are extracted through a specific Named Entity Recognizer. The concepts are queried

[23] https://lucene.apache.org/core/

to the knowledge base, which reasons about the importance of the tweet according to the keyword set provided by the user. This approach's evaluation seems inadequately small, using only 250 tweets from the 10.004 collected. Out of those 250, 60% were correctly identified by the knowledge base, 34% were completely incorrect, and the remainder we partially correct.

Ritter et al. [RIT15] describes how to use a small number of samples with an EM classifier to avoid manual classification. The EM model begins with ten to twenty positive samples and no negative samples. Ritter demonstrates that by training the EM only with positive events he achieves better results. Also, since EM does not require a large training corpus, it is simple to train various EM classifiers using a different seed for each. This approach was evaluated using 200 manually labelled samples from the training corpus. EM achieves better results than the other tested machine learning approaches, although it shows a difficult compromise between precision and recall; EM presents high precision rates but low recall (~90% - ~30%), and as the recall rate increases the precision rate decreases (~50% - ~50%).

Chang et al. [CHA16] show it is possible to improve Ritter et al.'s work using neural networks. Their architecture consists of word embeddings to model the tweets and Long Short Term Memory Networks to classify them. Chang et al.'s approach managed to increase Ritter et al.'s results in about 10% in both precision and recall.

Del Esposte et al. [ESP16] propose a recommender model to select OSINT relevant to the user's preferences. Nevertheless, this work was evaluated using a movie database and presents no suggestions for possible OSINT sources to use. The framework receives a keyword set from the user to find candidate information of interest to the user. The information selected by the framework is presented to be rated by the user. The ratings are used to train a recommendation model, which iteratively learns the user's preferences and tunes the model. Using this approach, Del Esposte et al. achieved a precision rate of about 70%, and on two different tests, recall rates of ~20% and ~7%.

3.1.2 OSINT collection and extraction methodologies

Another line of work presents methodologies for gathering and processing OSINT, which could then be applied in more specific contexts. Nunes et al. [NUN16] crawl some deep web's hacker forums and marketplaces. This approach goes directly to a major malicious user community, where state-of-the-art attacks and the latest discovered vulnerabilities can be found. Information gathered there could provide important forewarnings and enough time to patch vulnerable software. Nunes et al.'s approach begins by collecting web pages of vulnerability marketplaces and hacker forums. These pages are processed to obtain the textual contents discussed. The collected text is fed to an SVM that classifies it as security relevant or not, obtaining a precision and recall of 85% and 87%, respectively.

Mulwad et al. [MUL11], Neri et al. [NER09], and McNeil et al. [MCN13] process free text in search for security concepts. Mulwad et al.'s framework receives text snippets extracted from the web (e.g., blogs, news), which are classified by an SVM as containing security terms or not. The snippets classified as relevant are processed by a knowledge base that extracts the relevant concepts, such as the means of attack and the target of the attack. The extracted concepts are converted to the machine-readable OWL language format. This work was evaluated using NVD text excerpts describing vulnerabilities, testing if the framework could obtain the correct concepts from those texts. The framework identified 71% as containing security concepts, and the concepts detected were correct for roughly 90% of the cases.

Instead of presenting results, Neri et al. focus on describing the various correlation capabilities of their framework, which is widely used by governmental entities in Italy. The framework is composed of the following elements:

- A crawler that selectively collects documents from the Internet and databases;
- A lexical system that detects relevant concepts and the relation between those concepts;
- A search engine to query the collected knowledge;
- And a classification system that processes the query results, obtaining relations between the results, and assigning them themes.

McNeil et al. present a novel approach to detect cyber-security concepts from free text, called PACE. As described by McNeil et al., the typical bootstrap algorithm has a set of seeds that are used to search for patterns in the text to be processed, i.e., the algorithm has a set of initial sentences that are used as a pattern to find similar sentences. These algorithms require a large amount of seeds to obtain high recall percentages. Besides having seed patterns, PACE also has seeds of pairs (entity, context). These pairs provide flexibility to the pattern matching process, as a match can be found by either a seed pattern or by a seed context. This is especially useful for contexts where the terminology can vary greatly (e.g., application, software, program all refer to the same concept). PACE is evaluated using seeds manually extracted from ten cyber-security news articles, and by extracting entities from another seven. PACE obtained a high precision score (90%), but low recall (12%). Nevertheless, the authors compared PACE to the previous state-of-the-art method, which obtained zero results on the same dataset.

Erkal et al. [ERK15] also search for security concepts but use Twitter as data source. To avoid manually labelling a large dataset for supervised machine learning, they collect tweets from accounts focused on security news for positive samples, and tweets from generalist accounts (e.g., health, news) for negative samples. The tweets are processed using TF-IDF and classified as security relevant or not using Naive Bayes. Their approach is evaluated using cross-validation on the collected dataset, where the "percentage of correct decision" is 70%.

Jones et al. [JON15] created a framework for extracting concepts from free text. They use a bootstrap algorithm to extract entities from text using patterns. Their algorithm is based on the relation *(subject entity, predicate relation, object entity)* to extract concepts such as *(Microsoft, is vendor of, Internet Explorer)* from the sentence "Microsoft has released a fix for a critical bug that affected its Internet Explorer browser." A novel element is involving the user in the learning phase of the bootstrap algorithm. When the algorithm finds a new pattern to be included in its set of patterns, the user is queried for the correctness of the new pattern. To evaluate this work, the algorithm is trained with seeds originated from 62 security news posts; then, recall (24%) is calculated by running the algorithm on one manually labelled news post, and precision (82%) is calculated by manually verifying the correctness of the entities extracted from a corpus of 41 documents.

Liao et al. [LIA16] developed a framework for extracting IoC from scientific literature. IoC can be of many formats and are used to describe various aspects of an attack, such as the vector and the damage caused. Liao focuses on extracting IoC from technical literature since it possesses a more predictable structure, enabling high recall in this process. The text is processed by a complex pipeline composed of NLP processing tools. The terms are extracted and converted to the OpenIoC (addressed in Section 5) format, which can then be processed by automatic tools. Liao et al.'s tool presents a precision of 98% and recall of 93%.

Alqathani et al. [ALQ16] use the Apache MAVEN software repository[24] and NVD in their work. MAVEN is an open source software repository (primarily for Java) that simplifies dependency usage and software compiling; the libraries published there can be added as dependencies of any project using a simple mechanism. Alqathani's objective is to search the NVD for vulnerabilities in MAVEN's libraries. Then, following the MAVEN dependency tree they can identify which projects have those vulnerabilities. This work is evaluated by correctly identifying vulnerable projects. This approach's precision sits at roughly 90%, with an impressive recall rate of 100%.

3.1.3 Correlate user behaviour with OSINT for security

In this section we describe research work that processes user behaviour to infer malicious activity or vulnerabilities. Liu et al. [LIU15] try to predict if an infrastructure is vulnerable based on its observable behaviour and system configuration. First, they gather a set of system configurations; these include DNS, SMTP, and certificate management. Then, they gather observable behaviours of the system; they set up monitors outside the infrastructure and collect outgoing communications. The outbound traffic is analysed in search for spam messages, phishing attempts, botnet traffic, and scan traffic. These data are compared against a ground truth of three databases: the Veris Community Database,[25] the Hackmageddon,[26] and the Web Hacking Incidents Database,[27] all

[24] https://maven.apache.org/

[25] http://veriscommunity.net/index.html

[26] http://hackmageddon.com/

containing descriptions of real attacks. The databases' description of the victim and the means of attack are compared against the two descriptions collected from the user's system. Using a Random Forest classifier, they were able to predict if the infrastructure is vulnerable, with 88% of true positives and only 4% of false positives.

Miller et al. [MIL11] also use behaviour descriptors but for examining user behaviour in social networks. Miller et al. build a graph connecting the various elements of a social network based on their interactions and behavioural data. Through the graph, Miller et al. were able to discover threat networks, i.e., if a subset of elements in a social network present danger. Miller et al. test their approach using a dataset containing online social interactions, including the interactions of a terrorist group planning an attack. Their framework was tested using different parameters, and is able to show 100% precision rates while presenting a recall of ~60%.

3.1.4 Feed protection systems with OSINT

In this section, we describe research work that gathers OSINT and transforms it into a machine-readable format. This information can be fed to protection systems, such as IDSs or anti-viruses.

Mathews et al. [MAT12] and More et al. [MOR12] have the same objective: providing to an Intrusion Detection System information from traditional and non-traditional sources. As traditional sources they consider network data, sensors, and logs. Non-traditional data is comprised of information collected from online sources such as blog posts, news feeds or the NVD (i.e., OSINT).

Mathews et al. present a framework composed of an ontology, an IDS and a "Traffic Flow Classifier" (TFC). The ontology has three classes: means (the attack vector), consequences (the attack outcomes), and targets (the target system). The ontology is fed with OSINT data, gathered from various structured and unstructured sources. From OSINT, they extract three concepts corresponding to the ontology classes, which are used to update the ontology's reasoning capabilities. The TFC component monitors the packets' headers to infer if the traffic is legitimate or malicious based on traffic flow and used ports. The ontology uses a set of rules and the collected OSINT to process the data received from both the IDS and TFC, to detect attacks. This work was evaluated using a set of virtual machines generating benign and malicious traffic; the objective is to observe if the ontology correctly generates alerts for malicious traffic. Their best result is achieved with TFC using port, TTL and timing data, obtaining a true positive rate of 84%, and a false positive rate of 3%.

Zhu et al. [ZHU16] take a different approach. Their objective is to prove that it is possible to create an anti-malware solution using only information present in scientific literature. As described by Liao et al. [LIA16], scientific literature tends

[27] http://projects.webappsec.org/w/page/13246995/Web-Hacking-Incident-Database

to have a predictable structure, which simplifies NLP processing and content extraction. Zhu et al. process the scientific literature describing Android malware and extract features describing the attacks, creating a machine learning solution for recognizing malware. The results obtained are compared with manually engineered solution for the same purpose (Android malware detection). This work obtained similar results with a fully automated approach, using much less features, and with a system that can be easily updated. This work is evaluated on a few parameters, but we highlight that it achieves 92.5% true positive rate at 1% of false positives.

3.1.5 Gather exploit data from OSINT

Sabottke et al. [SAB15] use Twitter to gather mentions about existing exploits that are not yet present in security databases such as the NVD. This work shows that information about exploits is published on Twitter two days (in average) before these are included in the NVD. This work shows that analysing Twitter news streams can provide valuable data, as mentions about exploits can be seen there before they are formally recognized and normalized information is present in NVD.

Edkrantz et al. [EDK15] try to predict which vulnerabilities will have exploits. Although there are thousands of vulnerabilities described in the NVD database, only a small portion of those vulnerabilities have exploits created by hackers for malicious purposes. Therefore, Edkrantz et al. reason that it is possible to predict whether a vulnerability has an exploit based on its NVD descriptor. To test this methodology, they created a model using NVD's descriptions of vulnerabilities and their exploits. Then, the descriptions of new vulnerabilities are classified as likely to have an exploit or not. This approach presents both precision and recall around 80%.

3.1.6 Black-listed IPs

As mentioned in [SHA15], feeds are a good way to obtain information about external threats and with the use of OSINT it is possible to collect pertinent information from several feeds. A Blacklist is an example of a public list which contains information about threats and malicious behaviours.

There are some articles that investigate the effectiveness of blacklists and which blacklists, in a period, provide the most reliable information. Blacklists contain a significant number of false positives, as described in [KÜH14, ROS10, SIN08]. However, it is known that blacklist information is a widely-used measure for monitoring and detecting malicious behaviour [KÜH14, SIN08]. Four blacklists (NJABL, SORBS, SpamCop, and SpamHaus), which report suspicious email addresses considered as spam, were analysed in [SIN08]. It was used an unsolicited mail detection program for the confirmation and detection of false and true positives. After analysing email traffic in an academic environment (more than 7000 computers) within 10 days, the results confirmed that blacklists contain a high number of false positives.

The work done in [KÜH14] aims to understand the blacklists' contents and how its information is collected. The authors present two mechanisms: the detection of parked domains and the detection of sinkholes. They propose a mechanism to distinguish parked domains from benign domains, thus reducing a considerable number of non-benign domains present in a blacklist. A method for the detection of sinkholes it is also described, using a technique developed by the authors (graph-based), and their removal in blacklists. Sinkholes are, for example, servers that contain malicious domains, but have been controlled and mitigated by security organizations, which use them to monitor the network and communications with malicious domains. The authors conclude that blacklists only contain about 20% of malicious domains, resulting in a significant number of false positives.

In both previous works, it is difficult to correctly determine whether the effectiveness of a blacklist will increase or decrease over time.

AlienVault's OTX [ALI16] is a mechanism like the one being developed by EDP and FCiências.ID in DiSIEM (described in Section 4.1). This framework gathers information on IP addresses through denunciations by a set of communities. After this collection is obtained, the threat level of each of the suspicious addresses is assessed considering the number of attacks, the number of lists in which the address appears, and the type of maliciousness to which the suspected IP address is associated. The result is a list of IPs that can be used for monitoring or blocking IP addresses with a threat value calculated by OTX. However, the assessment is only made for OTX IPs and not for the blacklists chosen by the organization's security team.

3.1.7 Others

In this section, we describe research work that are singular in their objectives and do not fit any of the categories mentioned above. Kergl et al. [KER15] suggest a new response to anomalies or attacks mentioned on Twitter. Attack descriptions are collected from tweets, which are then compared to known attack descriptions collected from a vulnerability database. If the new attack's description matches a description present in the database, solutions to the vulnerability may have already been described; if not, it may be the sign of new zero-day vulnerability.

Zhang et al. [ZHA11] process the NVD to learn patterns between the characteristics of applications and their known vulnerabilities. Their objective is to use historical data about pieces of software and their vulnerabilities to train a model that predicts the time to next vulnerability, i.e., for a given piece of software, how long it takes until a new vulnerability is disclosed. Although Zhang et al. present an interesting idea, they were unable to generate a model presenting good correlation capabilities. As a vulnerability on a software version typically also affects the previous ones, various software versions are reported as containing the same the vulnerability. As different software versions that suffer from the same vulnerability (which is detected on a single day), are released in different dates, the authors could not obtain a model achieving good prediction accuracy.

3.2 Existing tools

Many tools are available that can be used to explore OSINT information, differing mainly on how they are delivered and on the features they provide. These tools may be categorised in three classes: generic open source tools; paid tools; paid services.

3.2.1 General purpose open source tools

Many general purpose open source tools may be used to collect, store and organize OSINT in general, but none of those found was designed specifically for security-related OSINT. For example, searching GitHub[28] using the "osint" keyword provides 324 results.[29] Nevertheless, most tools are either generic (i.e., collect OSINT from all sources such as the OSINT-Framework[30]) or collect only from a specific source, such as the various tools that collect Tweets.[31] These lack processing and analysis functionalities that would suit them for security-related OSINT applications, but can be adapted to a security context using two measures:

1. Focus the OSINT capture on security events or security-related sources;
2. Filter the data captured, keeping only security results.

The first measure reduces the scope of data to capture. For example, when using Twitter, one can select only security related accounts (e.g., Kaspersky[32]). The second measure reduces the amount of data to process by removing data unrelated to the context, whether filtering data not mentioning specific software elements or capturing only vulnerability data. Note that both measures can be used together.

To use these generic tools one is required to configure OSINT sources, to tailor existing analysis functions to the specific needs, or even implement required functionalities from scratch.

Table 3 presents two prominent examples of such open-source general purpose tools that provide mechanisms to implement both measures mentioned above.

3.2.2 OSINT paid tools

We searched for specialized tools for sale that can collect and process OSINT and that can be deployed within an organisation infrastructure. Most products found are sold as services, with the exception of Paterva's[33] Maltego product. Although

[28] https://github.com
[29] https://github.com/search?utf8=%E2%9C%93&q=osint&type=
[30] https://github.com/lockfale/OSINT-Framework
[31] https://dev.twitter.com/resources/twitter-libraries
[32] https://twitter.com/kaspersky
[33] https://www.paterva.com

Maltego is structured in a client-server way, where client applications are sold and Paterva's servers provide the services to customers, as an alternative, these servers can also be sold and deployed within an organisation network.

Tool name	Description
Logstash[34]	Logstash is one of the elements of the ElasticStack (together with ElasticSearch and Kibana – refer to Deliverable 2.1 "In-depth analysis of SIEMs extensibility" for more details). Logstash can collect data from a multitude of sources, including Twitter and other OSINT. In Logstash it is possible to place code to process and modify any of the data collected, which could be used as an initial filter for the data collected. After being processed, the data could be sent to ElasticSearch, where it is indexed. Finally, the data can be analysed using the several types of visualizations provided by Kibana.
IntelMQ[35]	IntelMQ is a message queue implemented to receive data from a wide variety of sources, including OSINT. IntelMQ's main feature is the collection and processing of security feeds (such as logs, tweets, or blacklists) autonomously. This tool enables the information security team to more efficiently collect information from a set of feeds. However, a setting is required for each source and, if the information we want to collect is different from the standard programs, it becomes necessary to create modules, or use similar modules, to correctly collect information from the intended source. Yet, as [SHA15] refers, it is necessary to gauge to which extent feeds are trustworthy and if indeed it is possible to rely on them, based on the information obtained to implement defence mechanisms.

Table 3 - Examples of general purpose open-source tools that can be extended for OSINT processing and analysis.

3.2.3 OSINT paid services

An alternative to using tools to collect OSINT is to purchase a security feed from specialized companies. These services are usually sold as a subscription to a feed of security-relevant news, sometimes specifically suited to the IT infrastructure of the subscriber. The main advantage of using one such feed is to simply pay for the data, instead of developing and managing another piece of software to gather such data. The main disadvantage is the cost of the service. Table 4 presents some security paid feeds we found, and their main features.

Service	Features
LookingGlass	Besides proving tools for threat analysis, also provides a set of specialized feeds: Threat Intelligence Services, a machine-readable feed, and Threat Mitigation
SecureWorks' Threat Intelligence	The provided descriptions are not very specific, but their service provides a threat intelligence feed tailored specifically to their clients.
Kaspersky's Threat Data Feed	Provides a feed consisting of rules mainly for botnet protection, and whitelist of legitimate services.

[34] https://www.elastic.co/products/logstash
[35] https://github.com/certtools/intelmq

Kaspersky's Tailored Threat Reporting	A security feed designed for a specific IT infrastructure. Details threat vectors, malware, attacks targeting specifically the IT infrastructure, possible information leaks, and a status on current attacks against the infrastructure.
RecordedFuture	Paid service for automated collection and visualization of security data focused on the costumer's needs. Includes support for the dark web, and a large set of visualization types.
FireEye cyber threat intelligence	Provides three services: • A subscription to a threat intelligence service, consisting of possible threats to an infrastructure, enriched with the attack's context • The support of a FireEye's security analyst • A service to use threat intelligence on the lifecycle of the company
Symantec DeepSight Intelligence	Depending on the subscription model, may include data about vulnerabilities, IP reputation, risk assessment; further, includes attacker data, such as active hacker campaigns and detected incidents.
Kenna vulnerability & risk intelligence platform	Integrates the results of vulnerability scan data with the results from 8 different threat feeds. Prioritizes vulnerabilities and provides risk reporting.
Airbus DS CyberSecurity cyber defence centres	Three European centres protect and monitor customer's assets in real time. Airbus DS CyberSecurity provides additional services for detection and investigation of sophisticated attacks, incident response, and risk analysis.
Anomali ThreatStream	Anomali ThreatStream is a Threat Intelligence Platform, allowing organizations to access intelligence feeds and integrate it with internal security and IT systems.

Table 4 - Examples of paid security feeds/services.

4 Preliminary results on OSINT processing

In this chapter we present the status and preliminary results of ongoing work in DiSIEM regarding the processing and analysis of OSINT, with the aim of creating tools that integrate relevant information into SIEMs. Detailed results and conclusions will be presented in a forthcoming deliverable (D4.2), as planned in the project Description of Action.

The work reported is related to the processing of two forms of OSINT: structured black-lists of IPs; and unstructured textual information arising from various kinds of sources. In the last case two main approaches are being followed: machine learning approaches to process posts of security-related twitter accounts; and DigitalMR's listening247 platform, which is employed for mining market trends, using data from various sources such as social networks, blogs and forums, to mention a few.

4.1 Blacklisted IPs OSINT processing

Blacklists are lists that contain information about untrusted elements and are a typical tool used as a cyber-defence mechanism [KÜH14]. An example of blacklists is a list of malware signatures, used by antivirus or Intrusion Prevention Systems (IPS). DiSIEM ongoing research focuses on IP blacklists, which are lists of IP addresses deemed as malicious; IPSs use them to block inbound and outbound connections to those IPs, which is a simple but effective security measure.

4.1.1 Trustworthy Blacklist in SIEM systems

EDP and FCiências.ID are working on a case study whose focus is on the trustworthiness of IP blacklists. One of the objectives of this ongoing work is the reduction of false positives when assessing the legitimacy of communications with IP addresses suspected of malicious activity.

To obtain a more reliable list of malicious IPs leading to a reduction of false positives it is necessary to classify the reputation of each IP address and each blacklist. This assessment is done using specific security metrics. Blacklists and their contents must be evaluated continuously (or whenever the lists change) and must consider the cases of communications from the organization's networks to blacklisted IP addresses.

Figure 2 represents an overview of the framework being developed, which includes four modules that can be used independently. The first is the IP Collector, a program with the purpose of gathering information from public blacklists. The second is the Trustworthiness assessment, which evaluates the reputation of the malicious IP addresses and the blacklists that contain them. The third module, the Trustworthy Assessment of Blacklists Interface (TABI) application, consists of a web management interface on the IP addresses, blacklists and cases related with communications between the organization and IP addresses suspicious of maliciousness. Finally, a reputable list of IPs

(BADIP.csv) is introduced in the SIEM and the rules for monitoring and generating alarms are defined. These components are described in the next sections.

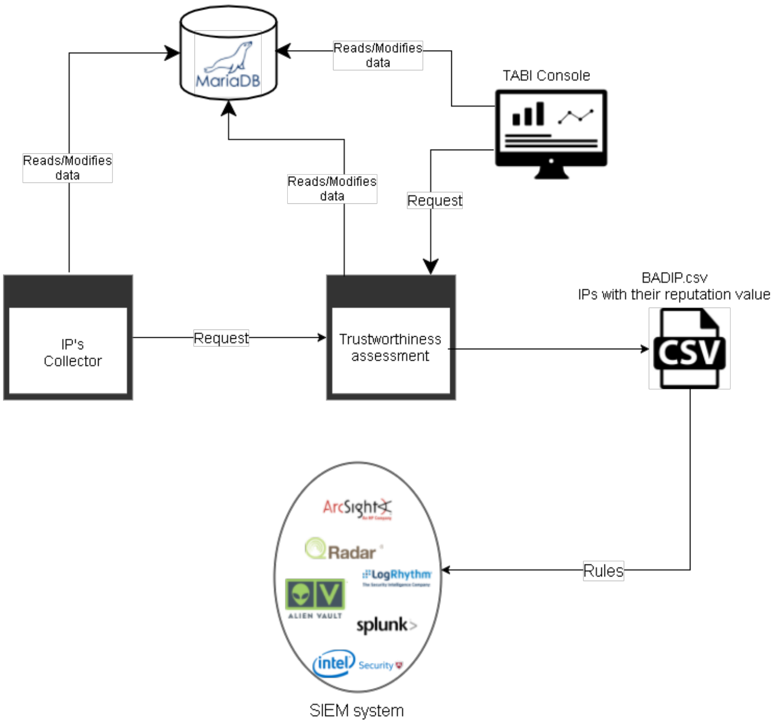

Figure 2 - Workflow of the IP blacklist processing framework.

4.1.2 IPs Collector

We consider as a source (or feed) an entity that provides one or more blacklists. In the undergoing study, we only consider public blacklists that contain information about IP addresses (IP blacklists). The framework uses the OSINT concept to gather information of a pre-specified set of public blacklists. At the end of a period of three months of investigations and selection of public blacklists, 28 sources and 121 blacklists were selected. This list of sources are included in Appendix A.

4.1.3 Trust Assessment

For an effective cyber-defence, and when there is an extensive number of IP addresses to consider, it becomes necessary to differentiate a suspicious IP address from another by its trustworthiness. The typical features used to differentiate them are the criticality, credibility, impact, maliciousness and the number of reports. The trust assessment aims to classify the reputation of maliciousness of an IP address and the reputation of credibility of a blacklist considering these conditions.

For the calculation of the reputation of maliciousness of an IP address, four features are used: Term Frequency (TF), precision, average of the reputation score of all blacklists that reported the IP, and its persistence. The IP rank is the position an IP occupies when sorting all IPs by the trust we have in their maliciousness, which is given by a reputation score.

The TF component is the relative frequency of one IP considering the number of occurrences of all gathered IPs. The precision of an IP is the ratio of the number of confirmed cases of malware detected by communications with this IP, to the total number of cases associated with communications with that IP in the current month. The persistence is defined for a given time period, which may be related with the SIEM event retention period, and is a measure of the IPs appearance in blacklists or in positive cases reported in the organization. As the solution should be adaptable to the environment of different organizations, when an IP has not been informed by blacklists, it is only discarded if it is not associated with positive cases.

4.1.4 Trustworthy Assessment Blacklists Interface

The Trustworthy Assessment Blacklist Interface (TABI) is a web interface, which is being developed to allow managing and visualizing information related with the blacklists, suspicious IPs, organization's cases and public organization's IPs. TABI will allow for a centralized management of the entire framework, without the need for code writing or file configuration.

The tool will consider the addition, removal and edition of blacklists and incident cases, to be used in the trustworthiness assessment of the IP addresses and blacklists. The TABI application will have an extra functionality that indicates if a public IP of the organization is contained in any blacklist. For this functionality to be operational, it is necessary to have access to a list of the public IP addresses of the organization.

4.2 Infrastructure-related OSINT processing

Another line of work in DiSIEM concerns the processing of unstructured textual OSINT that is posted on the web by cyber security companies and professionals, as well as hackers and attack victims. The information is posted on social networks such as Twitter, dedicated forums, blogs, and news feeds, to mention some of the publication venues.

4.2.1 Exploratory Machine Learning Approaches

We are exploring different machine learning-based approaches to discover relevant security-related OSINT for a given IT infrastructure. These approaches are oriented at keeping SOC analysts aware of the most relevant threats against the infrastructures under their responsibility, without requiring them to spend time searching for that information.

For this purpose, the DiSIEM industrial partners provided a description of the IT infrastructure they wish to monitor. This allows decreasing the amount of collected OSINT, therefore enabling the development of models tailored for the specific descriptions, and enabling also a more concise assessment of the performance of the approaches being followed and a more efficient infrastructure-aware OSINT discovery.

Collecting OSINT implies searching and collecting data from the most interesting sources. Nevertheless, security analysts have a limited time budget to seek this information, even though the quality of their work depends on this knowledge.

Our proposal is meant to provide analysts with the most recent and relevant information regarding the protected infrastructure. We want to maximize the amount of relevant information obtained, while minimizing the time required to view it. To achieve this objective, we propose a processing pipeline composed of an OSINT information gatherer, an automatic method for selecting the relevant information, and a summarizing function. More specifically, we use an automated tool to gather tweets from security-relevant accounts, and we are testing different machine learning approaches to select the relevant ones considering the protected infrastructure, and to group related information gathered to avoid presenting repeated or information.

The proposed methodology aims to simultaneously achieve three main objectives:

1. Maximize the amount of relevant information presented to the analyst;
2. Minimize the amount of irrelevant information presented to the analyst;
3. Aggregate related information.

The first objective aims to avoid discarding relevant information, while the second aims to avoid presenting irrelevant information to the analyst. These two objectives are fundamental to ensure the reliability of the system: analysts should trust that the presented information is relevant and must be taken seriously. The final objective is important to avoid the presentation of duplicate information.

Although there are many sources of OSINT, for these approaches we focus on Twitter for two main reasons. First, Twitter is well-recognized as an important source of short notices about web activity and about the occurrence of events in near real-time.[36] This is also true about cyber security events, as most security feeds and researchers maintain active accounts where they tweet the news' titles [CAM13, SAB15]. Therefore, Twitter is an interesting aggregator of information and activity from all kinds of sources. Secondly, since a tweet is limited to 140 characters (around 20-30 words), these messages are simpler to process automatically, enabling very high levels of accuracy and low false positive rates.

As agreed in the DiSIEM project proposal two types of machine learning approaches are being evaluated: well established methodologies such as SVM and Artificial Neural Networks (ANN), and deep learning approaches.

[36] https://www.americanpressinstitute.org/publications/reports/survey-research/how-people-use-twitter-in-general/

Support Vector Machines and Artificial Neural Networks. SVMs and ANNs are being tested to classify each tweet as relevant or not for a given IT infrastructure. Apache Spark,[37] a scalable platform, and its machine leaning library are being employed for this purpose. Figure 3 illustrates the proposed twitter classification architecture.

A data collector restricts tweets by collecting them only from relevant twitter accounts. Collected tweets are then passed by a group of filters that assigns them to a given part of the IT infrastructure. Then, a specific classifier is used to classify the tweets as relevant or not for the security of that part of the monitored infrastructure.

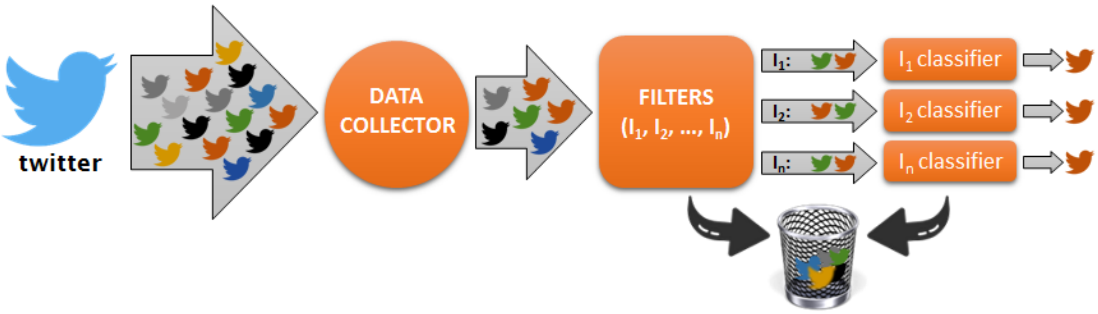

Figure 3 - Proposed Twitter classifier architecture.

Since Twitter is our data source and we want to avoid presenting retweets and the stream of similar tweets about the same events and threats, at the end of the architecture there is a clustering step used to group related tweets. Notice that at this level we are not interested in what the analyst will do with the relevant information, nor we aim to further process it to extract machine readable information (as is done by other works [LIA16, ZHU16]).

Ongoing work seeks to find good design parameters for SVMs and ANNs and to provide a comparison on the performance of these well-established machine learning techniques. Another interesting question being addressed is to find out if there is a clear benefit in using multiple classifiers for specific IT infrastructure parts instead of using a single classifier for the whole infrastructure. Preliminary results indicate that the objectives specified in the previous subsection may be met, but are still inconclusive regarding this question and also on the applicability of the methodologies to very large data sets.

Deep Learning approach. Besides SVMs and shallow ANNs, a deep learning methodology is being tested to classify each tweet as relevant or not for a given IT infrastructure. For this purpose, TensorFlow[38] is being employed.

Deep learning mechanisms have recently gained much warranted attention as they have been used in an increasing number of extremely complex tasks on very demanding big data problems. Regarding the problem at hand they are expected

[37] https://spark.apache.org/

[38] https://www.tensorflow.org/

to provide increased classification accuracy, less sensitiveness to the heterogeneity of data sources and data sets size, less requirements in prearranging a set of input features, and a higher level of generalisation and adaptation, thus increasing autonomy.

As such, the main objectives of following this approach consist in processing larger amounts of OSINT data with better classification accuracy as alternative simpler approaches.

We are interested in determining whether a tweet contains valuable information regarding a cyber-threat to a certain architectural component or not. This translates into a binary classification task: a tweet mentions a threat or not.

Figure 4 illustrates the proposed architecture for the neural network. We expect that a single deep learning model will present higher accuracy when classifying tweets for a complete IT infrastructure than using SVMs or ANNs. The input of the model is a sentence (a tweet) and a description of part of the IT infrastructure. The output should indicate if the sentence mentions a threat to that part of the infrastructure. Although the input mentions only a part of the IT infrastructure (as tweets generally mention one software element in their text), a single model will be used for the whole infrastructure.

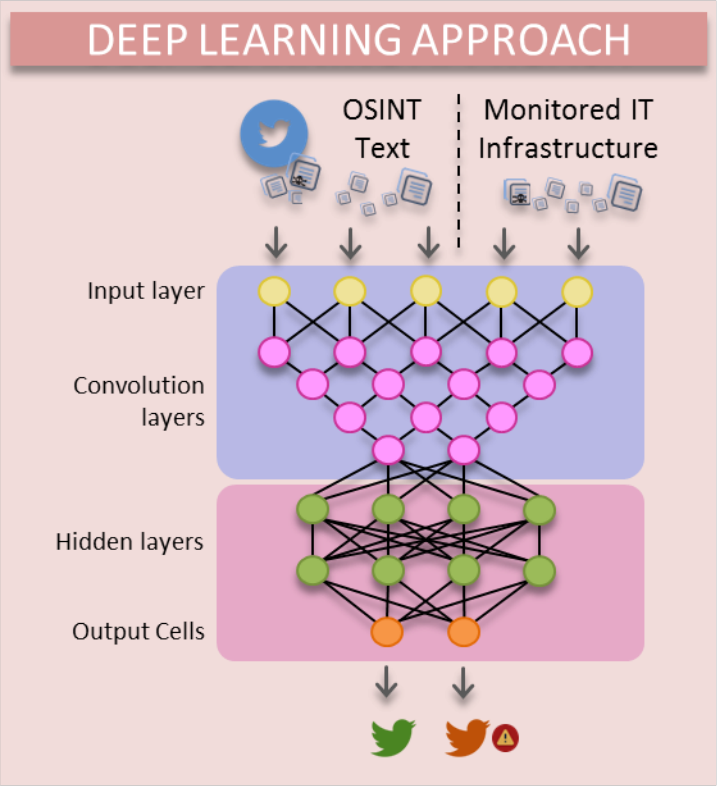

Figure 4 - Architecture of the deep learning approach to the classification of tweets.

Related work [KAL14, KIM14, WAN15] suggests neural network architectures where the input layer is a sentence comprised of concatenated word2vec [W2V] word embeddings, followed by a convolutional layer with multiple filters, a maxpooling layer, multiple fully connected layers, and finally a *softmax* classifier [CON17]. Therefore, we decided to employ a similar design. Figure 5 depicts the architecture being tested for tweet classification, which exploits the correlation between the tweet sentence and a specification of an IT infrastructure component by using a convolutional neural network. Preliminary results indicate that the approach is successful, although the dataset's size does not yet allow solid comparisons and conclusions.

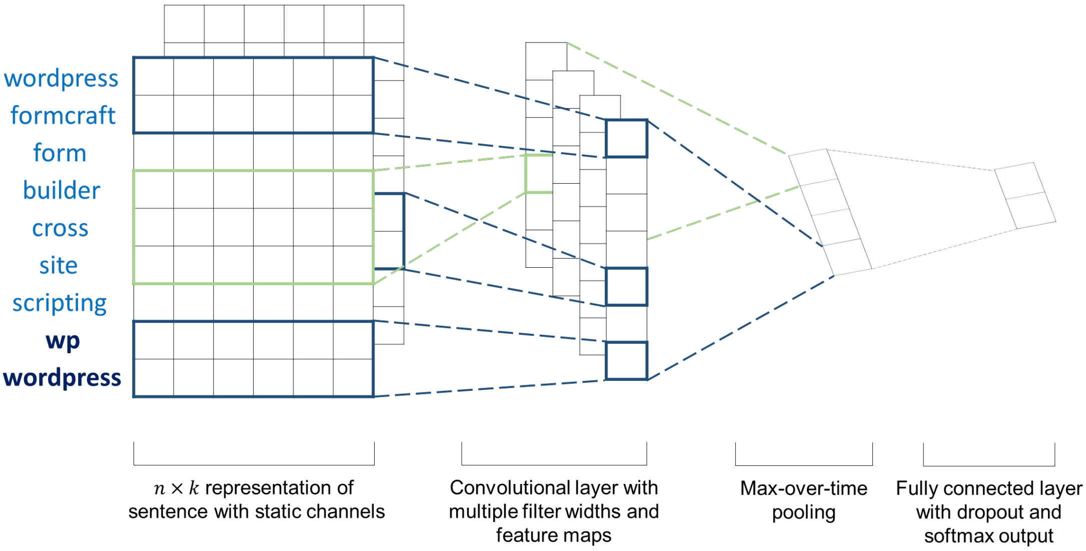
wordpress
formcraft
form
builder
cross
site
scripting
wp
wordpress
$n \times k$ representation of sentence with static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 5 - Deep neural network architecture for classification of tweets (adapted from [KIM14]).

4.2.2 DigitalMR Listening247 platform

The Listening247 platform is a service which offers analysis of various sources of data including blogs, social networks, news, boards/forums and other openly available data on the Internet for market research. It uses a Software as a Service (SaaS) model that enables users to monitor the web for specific subjects/topics while extracting insights and reports.

The platform has been designed for organisations to manage their reputation not only on social media, but various online locations. Such systems are increasingly in demand by senior marketing executives who look for ways to sift through fast-changing data across geographies, languages and time zones. This makes it particularly useful in this case as well as it provides a simple interface to process unstructured data.

Social listening helps organisations: accurately evaluate marketing campaigns; analyse hot conversation topics; discover white space/market gaps; respond to negative & leverage positive posts; benchmark your share of voice with competitors. Unlike conventional social media dashboards, this combined data from corporate CRM/ERP systems and millions of blogs, boards, videos and news from three different social media sites to present aggregated data quickly and clearly. In this case, analysis such as conversation topics can be useful for giving insight into unstructured data related to specific infrastructures of companies.

In terms of infrastructure, currently, the platform makes use of Amazon's cloud storage (S3), database services and elastic compute capability. This allows complex and processor intensive tasks to be offloaded, allowing them to utilise surplus processing and storage capability elsewhere in the cloud and concurrently with other tasks via APIs to multiple data aggregation engines to cover sources of online text such as Twitter, Facebook, blogs, boards, videos and news. The data is analysed to extract sentiment, brands, products and topics. This information retrieval step, combined with metadata coming from the online posts, is essential for the creation of insightful reports that describe what is said on the web about the subject of interest.

In terms of processing, the Listening247 is a distributed platform. It uses Hadoop MapReduce[39] for data processing and implements state-of-the-art machine learning algorithms for information retrieval. Raw and analysed data are stored in scalable distributed databases that offer a flexible query API used for our reporting needs. The back-end architecture is developed using Python, whereas the frontend is built using HTML5, JavaScript and PHP.

The main impact of the use of the Listening247 platform for DiSIEM is that it will be a better and more effective use of the social web data from both business and social perspectives. In particular, it will provide new insights/tools to better understand clients/citizens needs and activities that confine malicious content which will traditionally go unnoticed. Of course, personally identifiable information will be anonymised by any of the techniques that best fits this application in compliance with privacy laws.

Additionally, the platform has been used with success on several major languages including English, Spanish, German, Russian, Chinese, and Vietnamese among others. DigitalMR's network of 250 experienced and tested curators worldwide, in addition to our industrial strength processes for noise removal and disambiguating posts makes our training machine learning models stand out in terms of performance.

The proposed custom pipeline for the threat prediction based on the Listening247 platform will consist of a pre-processing step, noise filtering, and an analysis step where the entities of the tweet and the location (if available) can be extracted, and a prediction can be made as to the type of threat it is accompanied by a prediction confidence.

Inputs. The key inputs for the Listening247 will be keywords to focus the data being gathered from OSINT sources to only those relevant. This is the first step to noise filtering which involves forming queries that disambiguate keywords that might be homonyms to other words. For example, "Windows – the operating system" (an infrastructure), will yield information about "windows" which are used in buildings among others. Forming specialized queries for these relevant infrastructure as keywords such as for the Windows operating system will

[39] http://hadoop.apache.org/

narrow down the vast amounts of OSINT data available on the Internet, thereby making the amount of data to be processed more manageable. These queries can be updated with time to yield more relevant data.

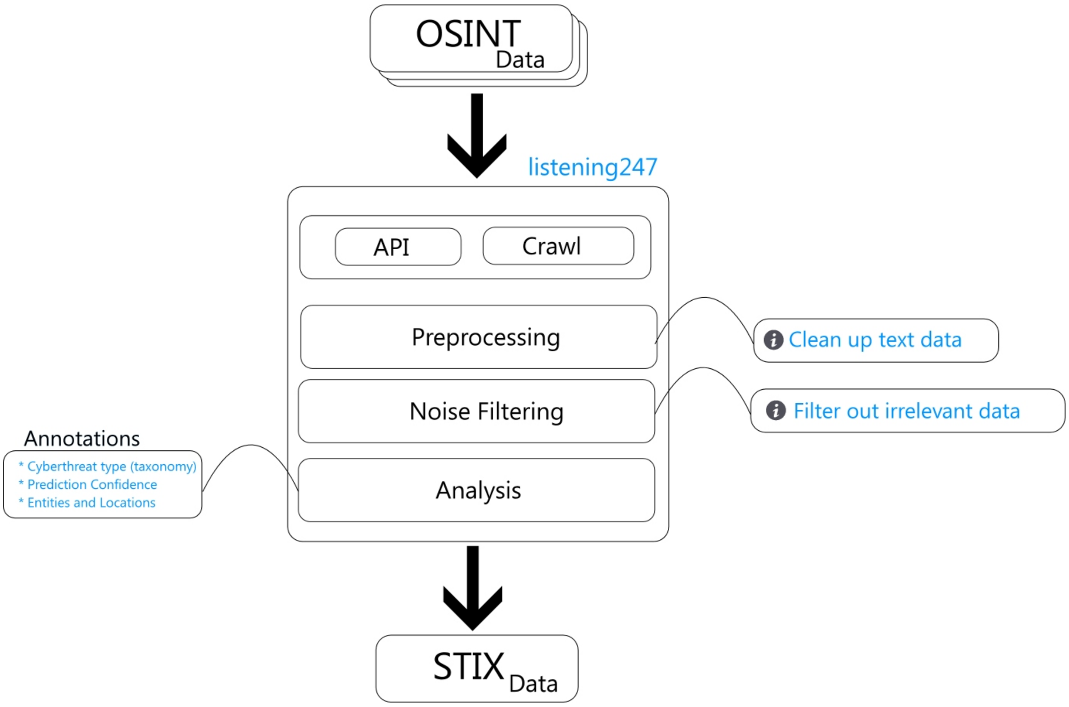

Figure 6 - DigitalMR OSINT threat predictor proposal.

Data Aggregation. To process all OSINT data, it needs to be aggregated with respect to time so that data occurring within the same period gets grouped together. This allows for time series analysis of the data from various sources, for example using Latent Dirichlet Allocation (LDA) to determine the topics of the data, or predicting the expected number of posts relevant to an infrastructure based on previously seen numbers for each week.

Information could be aggregated by week numbers (i.e. ISO Week format) which consists of 52, or 53 weeks in a year. This makes it more likely to have information related to each other within the same window (i.e. a week). Specifically, various sources of information have different velocities and aggregating information by week makes it more likely for information related to each other to be grouped together. By velocity, we are referring to the rate at which information is being produced. For example, when there is a Distributed Denial of Service (DDOS) attack on a website, social media sources like Twitter, Instagram and Facebook are usually the first ones to report the news. Followed by news agencies on their sites, and blog articles that follow up on the event. However, some of the velocities for these outlets such as news agencies are also changing due to the changes in the way news are being reported in this age of connectivity.

We can consider two designs for aggregating this data from various sources into a time-interval. The following provides two different approaches.

- Design #1

  In this design, the time interval is a day.

  ```
  {time-interval : (21-01-2017 , 22-01-2017) #daily
          {‘twitter’: <twitter_data>,   ‘blogs’: <blog_data>,
           ‘forums’: <forums_data> ….}
  }
  ```

  - Pros:
    - Data encapsulated within a common time frame.
  - Cons:
    - Data sources have different velocities;
    - Some data sources will be duplicates as a result.

- Design #2

  In this design, we account for the time difference of velocities for different sources which allows for similar information to hopefully be grouped within the same time-interval.

  ```
  { 'period':  (21-01-2017 , 28-01-2017) //one week data
          {'twitter':  {
                  'period':  (21-01-2017 , 22-01-2017) {
                          <twitter_data>
                  }
                  .
                  .
                  'period':  (27-01-2017 , 28-01-2017) {
                         <twitter_data>
                  }
          }//end of twitter data
          {'blogs': {
                  'period':  (21-01-2017 , 28-01-2017) {
                          <blog_data>
                  }
          }
  } //end of one week data
  ```

  - Pros:
    - Also encapsulates data within a time frame;
    - but also considers the velocity of the rate of capturing for each source.
  - Cons:
    - Rate of capturing for each source might need more optimizing.

We will be experimenting with various ways of aggregating data such that the most amount of relevant information (though at different velocities from different sources) are captured within the period.

Cyber threat Modelling. The first model in the pipeline filters out noise, specifically those data found to not be useful for the infrastructures of interest (see Figure 7). This model will need to be trained on a large number of OSINT data tagged with relevance to infrastructures of interest.

Filtered OSINT data will then be allowed to pass through to the second model (see Figure 8), which will use NLP to obtain meta-information such as the infrastructure involved and possibly the locations involved which will be added to the data in STIX v2.0 format (JSON).

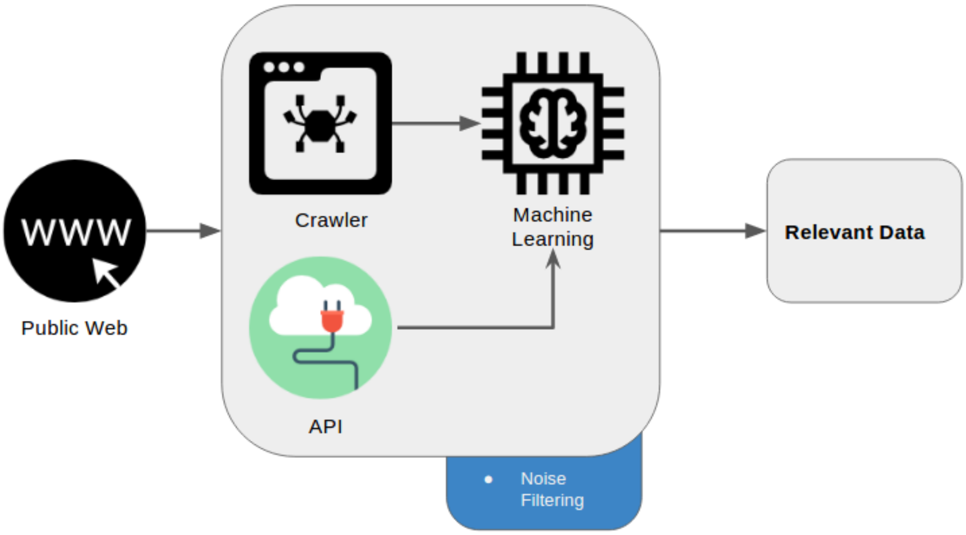

Figure 7 - Noise filtering step.

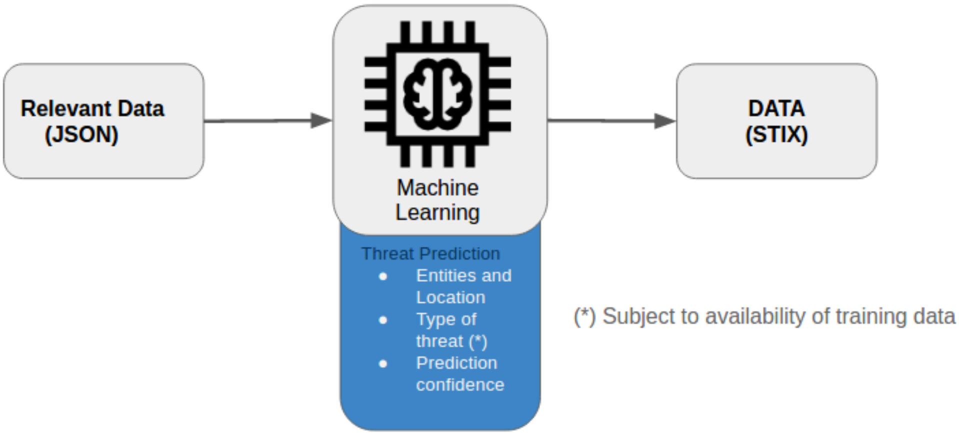

Figure 8 - Relevant data gets analysed by the machine learning and NLP tools.

Prediction of threat likelihood will require tagging data with threat for supervised training. So, the STIX payload could include information of threat likelihood to the infrastructure, and possibly a prediction confidence to avoid false alarms. This will likely involve the use of recurrent neural networks such as Long Short Term Memory (LSTM) neural networks or Gated Recurrent Neural Networks (GRU), which remember information over time and use that to influence their next prediction. This is essential for events that unfold over time.

LSTM networks contain a “memory unit”, which is selectively updated with new patterns. This memory is also used to selectively influence the neuron's final output. This selective process, which is handled by the gates, is learned over time. Figure 9 shows the canonical structure of an LSTM unit. They are generally trained by a back-propagation algorithm.

A potential restriction that will affect the accuracy of the classifier in predicting threats will be the availability of sufficient OSINT data relating to cyber threats. There is a need for getting annotated data in major languages that relate to cyber threats, which can then be used as training data for the threat predictor. LSTMs, in particular, require a lot of data because of the additional free-parameters. A potential solution could be the use of language processing to identify threats from the use of keywords that will typically indicate a threat in major languages; such as ‘ddos’, ‘security breach’, ‘leak’ and more. This data can be verified by humans and used as training data for the threat predictor. This implies that curators that are knowledgeable not only in the language, but also understand computer security will be needed. In addition to the type of threat, other information from the OSINT sources such as location and entities involved could also be extracted to provide a more comprehensive description of the threat. The prediction confidence of the classifier can be included in the data sent to SIEMS, which will help avoid the issue of false alarms.

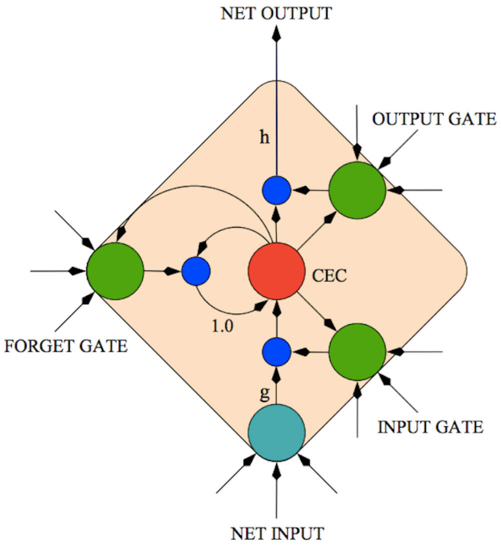

Figure 9 - An LSTM Neural Network Cell.

Other information that will be produced at this stage will include topics from time series topic modelling which helps pass on information about the topics discussed, how many posts are related to each of the topics and how this changes over time. Another feature is a word frequency count also done over time which captured the most frequent words for each period, and how significant they are relative to other words. All this data can be passed on to the visualization component which helps the end use of the SIEM have situational awareness of what is happening with regards to the OSINT data sources.

5 Context-aware OSINT integration

5.1 Threat Intelligence Data Interchange Formats

The number and impact of cyber-attacks has increased in recent years, as evidenced by reports from governments and organizations. To face these emerging threats, it is crucial to have timely access to relevant, sensitive threat intelligence information. Anyway, the ability to share this information is often not enough. Threat intelligence must be expressed and, then, shared using specific standards, allowing involved parties to speed up processing and analysis phases of received information, achieving interoperability among them.

Some companies developed their own application framework, in order to exchange cyber threat intelligence, relying on different standards and/or protocols. An example, is the one developed by Intel McAfee, called Open Data eXchange Layer (Open DXL) [MCA]. It supports a wide range of languages, allowing all the applications to communicate over a universal orchestration layer, and this interaction is totally independent of the underlying proprietary architecture. This abstraction from vendor-specific APIs makes the integration part much easier. It could represent a good solution for the integration with tools and platforms which already support Open DXL interchange methods, such as McAfee products, as possible future works.

However, in DiSIEM, as stated in [DIS], one of the architecture principles affirms that no additional or significant manual work should be required to operate and interact with SIEMs, as well as some relevant modifications due to our extensions. So, it is preferable to focus upon actual standards, instead of frameworks, to represent and exchange cyber threat intelligence, and check how it could be injected into SIEMs, using interchange methods which are already supported by them.

Starting from these considerations, some standards have been considered; the most important are the following:

- Incident Object Description and Exchange Format (IODEF) [DAN07]: XML based standard, mainly used for representing and sharing incident reports, especially when Computer Security Incident Response Teams (CSIRTs) are involved
- CyBOX/STIX/TAXII [CYB], [STR], [TRU1]: open standards developed by Mitre organization, respectively used for representing IoCs and detailed cyber threat intelligence, but also for sharing it with trusted partners
- OpenIOC [OPE]: vendor dependent XML based standard, introduced by Mandiant and primarily used in their product, however it can be extended in order to meet organization needs. It focuses especially upon tactical cyber threat intelligence, in fact it is not as complete as Mitre standards from this point of view, which are able to cover also strategic cyber threat intelligence in a more detailed way

Comparison among them was addressed in many articles and publications, such as in [KAM14], [FRA15], [FAR13], [SAU17]. Currently, we can state that the most used, and also the most promising, are the ones developed by Mitre organization, specifically Structured Threat Information eXpression (STIX), for describing cyber threat information, and Trusted Automated eXchange of Indicator Information (TAXII), for sharing it in an automated and secure way.

For these reasons, we decided on investigating more in details about these two standards, to understand if they could represent a good solution for DiSIEM objectives.

5.1.1 STIX

Structured Threat Information eXpression (STIX) is an open standard used for representing and exchanging Cyber Threat Intelligence (CTI), developed by MITRE organization, but now is maintained by OASIS Cyber Threat Intelligence Technical Committee. It allows sharing this information among different entities (e.g., organizations, governments, companies, research groups) in a consistent and machine-readable manner, with the aim of increasing:

- Collaborative threat analysis;
- Incident response capabilities;
- Automated threat sharing;
- Interoperability;
- Efficiency;
- Situational awareness.

It is widely used by many governments and organizations such as the National Council of Information Sharing and Analysis Center (ISAC council), Federal Government of the United States, U.S. Department of Homeland Security (DHS), Japanese Information-technology Promotion Agency (IPA), besides both commercial and government feed provides it, as well as many threat intelligence tools which are able to process and produce it.

Very briefly, it is targeted to support a large set of cyber threat management use cases, for example:

- Analysing cyber threats;
- Specifying indicator patterns for cyber threats;
- Managing cyber threat response activities;
  - Cyber threat prevention;
  - Cyber threat detection;
  - Cyber threat incident response;
- Sharing cyber threat information.

This standard allows binding together a diverse set of cyber threat information, which will be individually described later, representing it with a common standardized format. This is very important, especially when data should be fed into a Security Information and Event Management (SIEM) system. SIEMs are very powerful tools for empowering the organization security [DIS17], but they should work only with structured data, considering their limitations for ad-hoc importing and analysing unstructured formats.

This could be a problem, in fact, often, raw data extracted from external sources, such OSINT, Social Media Intelligence (SOCMINT), Human Intelligence (HUMINT), or other private or public repositories, are expressed through different data format (e.g., CSV, PDF, custom XML, custom JSON). This information is referred as Threat Data, and injecting it directly into SIEMs could led, for example, to a high number of false positives.

So, Threat Data should be collected, aggregated and, then, normalized, using a common structured format, before being analysed and enriched. The obtained cleaned data is referred as Threat Intelligence and it could be fed into SIEMs, to let these systems process and correlate it. These are some reasons that explain the importance of using standards for representing CTI, and, in particular, STIX is actually the most used.

Another great advantage of STIX is given by its extensibility. In fact, it is completely extensible, allowing each user to define their custom properties, custom objects or custom values for predefined properties. Obviously, it should be considered that some sharing parties could not be able to process a custom STIX file. In this case, they could choose to ignore the entire file or just the custom sections. However, if all the involved parties are aware of each additional data that could be present, there will not be any problem when processing it.

Besides being a widely used standard, lots of documentations can be found on the Internet. Many open-source libraries are available for helping developers to create and process CTI using STIX, written especially in Python, but also something in Java is available. Both libraries and documentations are continuously updated; currently the latest stable version is the 2.0.

Next sections will proceed with a more detailed description about the current version of STIX (2.0). Finally, a brief comparison among this version and the older ones will be made, for pointing out why it could represent a good choice for representing CTI in DiSIEM.

STIX 2.0. There are many differences among this version and the others. STIX 2.0 [OAS17], [OAS171], [OAS172], [OAS173] could be considered graph-based, where nodes and edges are respectively STIX Domain Objects (SDO) and STIX relationships (that could be STIX Relationship Objects (SRO) or embedded relationships).

STIX Domain Objects are extensions of STIX 1.x [STR] core constructs, while STIX Relationship Objects indicate explicit relationships among different objects, allowing to represent in a more understandable way the related threat intelligence. Before starting to explain the available SDOs, it should be considered that CybOX standard has been integrated into STIX 2.0, to describe simple IoCs and their associated patterns. The predefined SDOs are the following:

- **Observed Data:** related to basic Indicator of Compromise, such as IP addresses, hash-files and registry key values. These STIX Objects are used to represent what has been monitored. Malicious activities are recognized checking them with patterns described in STIX Indicators, where detailed information about the threat is provided;
- **Indicator:** describes patterns used to detect malicious or suspicious cyber activities. These patterns could be specified using the STIX Patterning Language [OAS174];
- **Identity:** represents individuals, organizations, groups, but also classes of individuals, organizations and groups. It could refer both to attackers and victims;
- **Attack Pattern:** type of TTP for categorising attacks, generalizing them to the pattern that they follow, providing detailed information about how they are performed. Reference to externally-defined taxonomy, such as CAPEC [CAP], could be attached to this SDO;
- **Malware:** refers to malicious code and/or software. This object helps to characterise, identify and categorising malware samples through a text description field;
- **Campaign:** describes a set of malicious activities performed by an attacker to a specific Identity over a specific period of time;
- **Intrusion Set:** set of Campaigns, performed by an attacker, targeted to a specific resource of a specific Identity;
- **Course of Action:** countermeasures to be taken against a specific threat, in order to mitigate the possible impacts of incidents;
- **Threat Actor:** represents malicious actor identity, including his historical observed behaviour against a specific entity;
- **Tool:** legitimate software that can be used by adversaries to perform attacks. Examples could be remote access tools (e.g., RDP) and network access tools (e.g., NMAP);
- **Vulnerability:** refers to mistakes in software that can be exploited by attackers. External references (e.g., CVEs) could be attached to the SDO;
- **Report:** collections of threat intelligence related to one or more topic (e.g., malware, attack technique, threat actor).

Instead, the predefined SROs are the following:

- Relationship: used for linking two SDOs in order to explicitly defined how they are related to each other. They can be considered as edges in a hypothetical graph, where SDOs are the vertices.
- Sighting: refers to the belief that something in CTI was seen (e.g., indicators, malware, observed data). Used for tracking threat actors, resources targeted, suspicious behaviours, etc.

STIX 2.0 objects are completely customizable. There are two primary means of customization:

- Custom Properties: for adding not already defined properties to SDOs
- Custom Objects: for creating from scratch new SDOs

In order to perform these operations, some specific rules should be followed, regarding naming, length, what ASCII character should be used, etc.

Additionally, some SDOs contain a particular property where the set of possible values that can be assigned to them, is associated to an open vocabulary. It means that these set of values could be seen as a sort of "suggested values", but, in practice, any other values could be used.

Differently from STIX 1.x, this version exploits JSON standard to represent STIX objects (it is for this reason that STIX Objects are considered for this version, not STIX files, as in the previous ones), instead of XML. OASIS CTI Technical Committee (TC) stated that JSON was more lightweight than XML, and sufficient to express the semantic of cyber threat intelligence. Besides, it is simpler to use and globally preferred by developers. Some open-source utilities and libraries can be downloaded from the website, for creating and processing STIX 2.0 objects.

Another important feature of STIX 2.0 is that it is completely transport-agnostic. It means that it does not rely on any specific transport mechanism, and this is achieved embedding the STIX Objects that should be sent into a Bundle, provided by STIX, that can be seen as a sort of container for STIX Objects.

For more detailed information and some practical example, the documentation available in the website can be consulted.

Comparison between STIX 1.x and STIX 2.0. In this section, a brief comparison among the versions will be considered, to understand why STIX 2.0 would be a better choice with respect to STIX 1.x, for DiSIEM project:

- It is more recent. It seems trivial, but being more recent, more efforts will be spent in order to update and improve it, considering the high number of differences than previous versions;
- JSON vs XML. As explained in the previous section, JSON is more lightweight, simpler to use and preferred by developers;
- One standard. CybOX Standard is completely integrated in STIX 2.0, while in STIX 1.x it is a separated standard;
- STIX Domain Objects: in STIX 1.x, Objects are embedded into each other, while in STIX 2.0 they are defined at the top level, and the relationships among them are expressed through SROs. Besides some STIX 1.x construct were split, in order to generate different and more detailed STIX 2.0 Object. For example, STIX 1.x TTP construct was split in STIX 2.0 Attack Patterns, Tool, Malware and Vulnerability Objects;
- Introduction of SROs as top level objects;
- Data markings don't use anymore a serialization specific language, such as XPath. In STIX 2.0, markings could be applied to entire objects or to specific parts of them;

- Indicator Pattern Language. In STIX 1.x, indicator patterns were expressed in XML, making very difficult to express complex patterns. While, in STIX 2.0 specific indicator pattern languages can be used, independent from the serialization language, making them easier to read, process and for describing more complex situations.

The only (temporary) advantage of using STIX 1.x, is related to available on-line documentation, open-source utilities and available samples. However, this advantage will disappear once new updates will be available for STIX 2.0, and, considering the importance of the topic and how much this standard is actually used by many governments, companies and organizations, it should happen very frequently. Anyway, actual available STIX 2.0 libraries are good enough to create and process STIX JSON objects, and considering the above differences among considered versions, it can be stated that, for normalizing unstructured cyber threat data gathered from external sources, to structured data to be injected into SIEMs, STIX 2.0 will be a better choice than previous versions, in the context of DiSIEM project.

5.1.2 TAXII

Trusted Automated eXchange of Indicator Information (TAXII) [TRU1] is an application layer protocol, developed by MITRE organization for communication of CTI in a simple, automated and scalable manner. It was led by the DHS and facilitated by MITRE organization considering STIX standard, in fact it must support the exchange of STIX content; this feature is mandatory to implement. However, additional content types are permitted. Now it is maintained by the OASIS Cyber Threat Intelligence Technical Committee, the same as STIX.

Thanks to its high level of interoperability with STIX itself, it is widely used by many organizations and also governments. Some examples are the Advanced Cyber Defence Center (ACDC), the ISAC Council and IBM for its cloud-based platform IBM X-Force Exchange.

TAXII goals can be summarized as follows:

- Enable timely and secure sharing of CTI in cyber sharing communities;
- Support a broad range of use cases and practises common to cyber information sharing scenarios;
- Minimize operational changes needed to adopt TAXII.

Anyway, it is important to point out that this standard does not allow defining trust agreements between sharing partners, as any access control limitations or non-technical aspects of cyber threat information sharing. Instead, it enables parties to share situational awareness, basing on already existing data and trust sharing agreements.

It enables secure sharing of CTI, considering that it is a transport mechanism built over HTTPS. Besides, it supports many common sharing models, such as hub and spoke, publisher/subscriber and peer-to-peer, so, it is suitable both for centralized and decentralized environments.

Next section will proceed describing the newest version, TAXII 2.0. Due to the choice of using STIX 2.0, we decided on focusing directly on this version of TAXII, and to not consider the previous ones. The main reason was that TAXII versions were specifically developed considering the related STIX version, in order to exploit all its potentialities, although they could be used for transporting other content types.

As stated in the previous section, there are huge differences between STIX 1.x and STIX 2.0, starting from the format used for representing cyber threat intelligence, so, we decided on focusing upon last version of TAXII, for checking its applicability in DiSIEM context.

TAXII 2.0. With STIX 2.0, CTI started to be represented using JSON instead of XML. For this reason, also TAXII had to be modified, in order to deal with JSON as main standard used for representing CTI. Previous versions, in fact, were developed for, mainly, dealing with STIX files expressed through XML format. Detailed specifications of this version could be found in [OAS175].

The support for exchanging STIX 2.0 content is mandatory to implement, anyway additional content types are permitted. It is designed to work specifically with HTTPS, to enable secure and authenticated communication between sharing parties. Actual specification does not define any requirements for HTTP.

This standard defines two primary services for supporting most common sharing models, both for centralized and decentralized environments:

- **Collections:** producers (TAXII Servers) can host a set of CTI that can be, in turn, requested by consumers (TAXII Clients). Information is exchanged in a request-response manner.
- **Channels:** used by producers to push data to different consumers, and by consumers to receive data from producers. Channels are well suited for publish/subscribe sharing models, where consumers perform subscription operations over producers, to receive specific CTI.

Channels, Collections and related functionalities can be grouped together to form an API Root (TAXII Servers could host many API Roots), allowing a division of content and access control rules by trust groups or any other kind of grouping. A simple example, to better understand this concept, is given by a TAXII Server, which could host two API Roots, one used by “Trust Group A” and the other by “Trust Group B”.

TAXII 2.0 defines two ways for allowing TAXII Clients to identify TAXII Servers. The first is a network level discovery, which allows the latter to advertise their location within a network. The second, instead, uses a Discovery Endpoint, which identifies an URL and an HTTP method with a defined request and response, for enabling authorized clients to gather information about the server.

Authentication and authorization is implemented as defined in [FIE14], through the Authorization and WWW-Authenticate HTTP Header respectively. HTTP Basic Authentication [RES15] is the mandatory schema to implement in TAXII 2.0. Anyway, other authentication schemes can also be supported.

Another important feature is related to the possibility of customizing this standard, adding new properties, in order to improve information exchange. Specific naming conventions should be followed for every custom property, being careful to let all the involved parties aware of these modifications, to avoid processing problems when this data is received.

TAXII in DiSIEM context. After a brief TAXII 2.0 description, in this section will be inferred if its usage could be a valuable addition for DiSIEM project.

In the previous section, it has been stated that TAXII 2.0 is the actual standard used for exchanging cyber threat intelligence represented using STIX 2.0 standard. This is actually true but, however, there are still some disadvantages, for how concern its usage, especially considering DiSIEM context.

An important drawback regards the lack of available open source libraries and utilities, for helping developers to implement TAXII Clients and Servers. We can state that TAXII 2.0 is not mature enough, differently from STIX 2.0, from this point of view. Besides, SIEMs do not support yet TAXII 2.0 protocol, and one of the DiSIEM architecture principles [DIS] affirms that they should not be modified due to our extensions and no additional or significant manual work should be required to operate with them.

So, considering the high number of mandatory specifications to implement for building a TAXII 2.0 service from scratch, without the possibility of relying on external open source libraries and utilities, we decided on thinking about alternative solutions for exchanging STIX data, in order to inject the output of DiSIEM OSINT-based components into SIEMs.

In conclusion, STIX data are expressed through JSON format, so, for DiSIEM use cases, it could be better to consider interchange methods already supported by SIEMs (e.g., Syslog, LogStash), mentioned more in details in the Integration Plan related to [DIS], which support JSON ingestion.

5.2 Integrating OSINT data with infrastructure events

In previous chapters, different approaches for OSINT data fusion and analysis have been described. The final objective is to integrate the relevant security data coming from these public sources with data gathered from the infrastructure by the SIEMs to anticipate and improve the threat detection.

In this context, it arises the need of a component that considers what is happening inside the monitored infrastructure providing a threat score for incoming OSINT data that helps to identify its relevance and priority. This threat score will complement the usage of static information about the monitored infrastructure with dynamic and real-time threat intelligence data reported from inside the own monitored infrastructure in the way of IoCs.

Entering more in detail, we need a component able to perform this correlation considering static and real-time information, such as Indicators of Compromise, related to the monitored infrastructure and data coming from OSINT sources through other DISIEM OSINT data fusion and analysis tools, for checking the relevance of the latter flow depending on the former. This dynamic evaluation will be based on heuristic analysis which allows determining the priority of the incoming OSINT data, assigning a threat score to it. The details about how the score is calculated with this particular method will be explained in the remaining of this chapter.

The final STIX object integrating the information received from OSINT data sources with its calculated threat score for the infrastructure, can be sent directly to the SIEMs for its visualization, storage or processing, or be sent back to the DiSIEM OSINT-based components as a feedback, in order to refine the machine learning algorithms with relevant information based on real-time analysis, improving threat detection and prediction. This will allow achieving a context-aware OSINT data analysis.

5.2.1 Architecture proposal

The proposed architecture, depicted in Figure 10, is composed of the following modules: (i) the *Entry Point*, which obtains information coming from multiple sources (e.g., OSINT data, infrastructure, IoCs, etc.), to be used in the threat score analysis performed by the heuristics engine; (ii) the *Database* that will store the information of the infrastructure and the OSINT data collector; (iii) the *Heuristics Engine*, which will compute a threat score based on the information received from the infrastructure and the OSINT data collector; (iv) the *Threat Score Agent*, that will build the final IoC object with the obtained result and will share it and interact with the SIEMs.

Entry Point: this module will be responsible of capturing useful data from OSINT, IoCs and the infrastructure in order to evaluate the set of pre-defined heuristics and to compute a threat score. The entry point will separate the input data into two main groups: infrastructure data and OSINT data. The former needs to be stored in the database, whereas the latter could be directly used by the engine without storing it.

Heuristics Engine: It will be mainly responsible of using the input data (e.g., context information, features) coming from the infrastructure in the analysis process. The latter considers a set of conditions that are evaluated for every single feature. A score (either positive or negative) is assigned to every feature (i.e., individual score). The sum of all individual scores results into the Threat Score associated to the data being analysed.

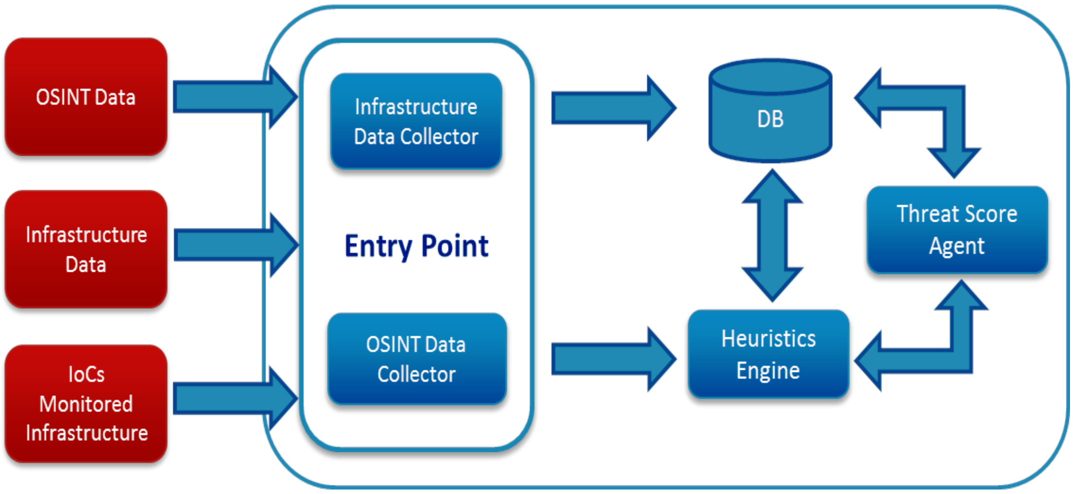

Figure 10 - context-aware OSINT Data Architecture

Database: Information received from the infrastructure needs to be stored in a database to be used later by the engine or the agent. The data received from OSINT sources that do not need to be stored in the database can be immediately sent to the Heuristic Engine for their analysis.

Threat Score Agent: It will be mainly responsible for the generation of the resulting Indicator of Compromise, including the threat score for security information received from OSINT data sources. This IoC that will be shared by this component would include the same information received from OSINT (JSON following STIX format) but adding the threat score as well as the features considered in the evaluation. This module will provide the interfaces used to interact with the SIEMs or any other component interested in this score and other useful information related to it. In addition, this module will interact directly with the database, when to retrieve additional information related to the heuristic or the data being analysed.

5.3 Context-aware Threat Score

5.3.1 Heuristics-based threat score

The heuristics-based threat score is composed of a set of individual scores that could be used in complement with other DiSIEM prediction tools to indicate the priority and relevance for the infrastructure of incoming security information received from OSINT data sources. Different aggregation techniques can be used for the computation of the Threat Score from the simplest way performed as the sum of all individual scores (score assigned to each heuristic) using following Equation 1, to more sophisticated ones using ordered weighted averaging (OWA) operators [Jos12] or Weighted Ordered Weighted Aggregation (WOWA) operators [Ern06].

$$Threat_{Score} = \sum_{i=1}^{n} Score_i \quad (1)$$

Depending on the information that is available from both, the infrastructure and the threat intelligence received from OSINT data source, it will be analysed the best aggregation method for calculation of the final threat score.

Considering, for instance, that one of the features to be evaluated is the presence of a Common Vulnerability Exposure – CVE [MIT] – identifier in the input data, the engine will check if the word 'CVE' appears in the input data in order to retrieve the complete CVE number composed of the publication year and the identification number (i.e., CVE-AAAA-NNNN). If a CVE is found, the engine then checks for its associated Common Vulnerability Scoring System (CVSS) [FOR17], more specifically, the engine will search for its associated base score, which considers access vector, access complexity, authentication, and impact related information based on availability, confidentiality and integrity. Depending on the CVSS score, the vulnerability is labelled as none, low, medium, high or critical, as shown in Table 5.

Severity	None	Low	Medium	High	Critical
Lower bound	0.0	0.1	4.0	7.0	9.0
Upper bound	0.0	3.9	6.9	8.9	10.0

(Source: https://www.first.org/cvss/specification-document)

Table 5 - CVSS v3 Ratings.

Each evaluated feature is assigned an individual score based on the defined threshold (e.g., from 0 to 5) that will indicate the level of impact of the feature with respect to the event. The following example illustrates this assignment.

We define the variable "Score_CVE" that will compute the individual score value assigned to the presence of a CVE in the input data based on the conditions described in Table 6.

Other features (e.g., source IP, created by, valid until, etc.) may use positive and/or negative values in the assignment process. Such individual values are then tuned in the training and calibration processes so that the final threat score reduces the number of false positives and negatives.

Evaluation	Condition	Score_CVE
Evaluate if there is not a CVE in the input data	If CVE == ' '	0
Evaluate if there is CVE in the input data with CVSS = 'none' or 0.0	If CVE != ' ' & CVSS = ´none´ | CVSS = 0.0	1
Evaluate if there is CVE in the input data with CVSS = 'low' or less than 4.0	If CVE != ' ' & CVSS = ´low´ | CVSS < 4.0	2
Evaluate if there is CVE in the input data with CVSS = 'medium' or less than 7.0	If CVE != ' ' & CVSS = ´medium´ | CVSS < 7.0	3
Evaluate if there is CVE in the input data with CVSS = 'high or less than 9.0	If CVE != ' ' & CVSS = ´high´ | CVSS < 9.0	4
Evaluate if there is CVE in the input data with CVSS = 'critical' or less than 10.0	If CVE != ' ' & CVSS = ´critical´ | CVSS < 10.0	5

Table 6 - Individual Threat Score.

5.3.2 Threat Score Methodology

The threat score evaluation uses a heuristic analysis methodology composed of the following steps:

1. **Source Identification:** Input data may come from different sources, therefore, during this phase we need to search and identify all possible sources of information for our tool. Examples of these sources are: security logs, databases, report data, OSINT data sources, IoCs, etc.

2. **Heuristics Identification:** Different features (e.g., heuristics) can be identified from the input data. Such features must provide relevant information about the infrastructure (e.g., vulnerabilities, events, faults, errors, etc.) that could be useful in the threat analysis and classification process. Examples of heuristics are: CVE, IP source, IP destination, port source, port destination, timestamp, etc.

3. **Threshold Definition:** For each heuristic, we need to define minimum and maximum values that could be assigned based on characteristics associated to the instance. We can check, for instance, if the input data contains or not a CVE for the detected threat. A threshold of (e.g., 0–5) can be assigned to cover all possible results described in Table 6 - .

4. **Score Computation:** For each possible instance of the identified heuristic, a score value is assigned based on expert knowledge. Scores associated to each heuristic can be either positive or negative, depending on the impact of the selected heuristic. All individual scores are then summed up and a final score is computed. The resulting value will indicate the priority and relevance of the security information coming from OSINT data sources for the monitored infrastructure.

5. **Training Period:** We need to perform a set of preliminary tests (during a training process) to evaluate the performance of the engine. The tests should include real data so that we can analyse the score obtained individually (for each heuristic) and globally (for the whole event). The higher the number of tests during this phase, the better for the tool to evaluate false positives and/or negatives, and to avoid deviations.

6. **Engine Calibration:** Since preliminary tests are based on the assessment made by expert knowledge, we need to minimize deviations (e.g., reduce number of false positive, false negative) by analysing the obtained results, adding other heuristics and/or modifying the assigned values to current attributes. It is possible that during the training process, we realize that instead of giving a value of 3 or 5 to CVEs with medium and high impact base scores we should give a value of 2 and 3 respectively.

7. **Final Tests:** Once the engine is calibrated, we can repeat previous tests or add new ones in order to evaluate the performance of our tool.

5.3.3 Preliminary analysis of heuristics features

The two main inputs of our context-aware OSINT data analyser are the following:

- Security information coming from OSINT data-sources provided by DigitalMR platform and/or FCiências.ID OSINT-based component.
- IoCs coming from the monitored infrastructure.

Several features coming from each of those sources can be considered in the threat score evaluation. Similar features can be merged into a more enriched group of heuristics and a sub-score could also be assigned to the group so that its impact can be analysed accordingly. Some examples of potential features to be considered by our context-aware OSINT data analyser are described as follows:

- External references to non-STIX information, used for a better description of the threat, such as CVE for vulnerabilities and CAPEC for attack patterns
- Validity time interval of the STIX Object
- Identity of threat actors or entity who are actually being targeted
- IPs or domain names included in IoCs and reported as source of some incident detected in the infrastructure
- Type of the activity related to a particular IoC (e.g., malicious, anomalous, benign)
- Phase of the kill chain where the threat was detected

These features, and others, will be stored and used by the Heuristic Engine when requested the assignment of a threat score to a specific IoC. New dynamic features could be added in the future to the heuristic analysis to improve the evaluation performed.

As described in Section 5.1, in DiSIEM we are going to consider that all the incoming threat information is expressed through the STIX 2.0 standard, in JSON format, and this assumption is valid for both the input flows.

To perform the threat score assignment, and based on the aforementioned input data, we decided to start the analysis focusing on the following dynamic features from the SDOs defined in STIX 2.0:

- **Indicators:** used for detecting malicious activities;
- **Vulnerabilities:** detailed information about known vulnerabilities which are related to some relevant assets of the monitored infrastructure;
- **Attack Patterns:** used for describing specific properties related to various attacks;
- **Tools:** information about tools that could be used for performing a specific attack;
- **Threat Actors:** information about malicious entities who are behind specific malicious activities;

- **Identities:** detailed personal information about both malicious and not malicious entities.

The selection of this initial set of objects can be later extended to more complex STIX Domain Objects such as Intrusion Set, Campaign and Report Objects in case some of them are relevant for the DiSIEM validation use cases or pilot deployments. Malware and Course of Actions Objects have not been considered by the moment because in the actual STIX version 2.0 they are still stubs.

Concerning Observed Data Object, they could be sent by a SIEM with data related to what it is monitoring and be used for matching specific patterns expressed in STIX Indicators received from OSINT data sources. However, this would translate into the implementation of threat detection capabilities in this component which is not its purpose. This operation is out of the scope of this component, because it should deal with intelligence ready to be used, therefore, this kind of STIX Object is not considered in the threat score evaluation.

Coming back to SDOs that, instead, will be used for the threat score assignment, we had also made a preliminary identification of some features that could be interesting to be used for the heuristics analysis. Different SDOs actually share common properties, so we started studying these ones. Then, we proceeded to consider also specific properties for each of these objects.

Common properties: SDOs are characterised by some common properties, which are used for describing related features. The initial set of them that will be taken in consideration is composed by the following ones:

- **Type:** indicates the type of the SDO (e.g., indicator, vulnerabilities, tools);
- **Created:** timestamp that indicated when the SDO was created;
- **Modified:** timestamp that indicated the last update made on the SDO;
- **Revoked**: timestamp that indicated when the SDO was revoked;
- **External_references:** refers to non-STIX information, used for a more accurate description of the object (e.g., CVE for vulnerabilities, CAPEC for attack patterns).

Indicator: some interesting specific properties of this SDO are:

- **Labels:** open-vocabulary field that indicates the kind of the detected activity. Possible values could be "anomalous-activity", "malicious-activity", "anonymization" and "benign;"
- **Pattern:** detection pattern for this indicator. For example, it could contain a set of malicious IP addresses or domain names;
- **Valid_from:** time from which this object should be considered valuable intelligence;
- **Valid_from_precision:** precision of the previous timestamp;
- **Valid_until:** time until which this object should be considered a valuable intelligence;
- **Valid_until_precision:** precision of the previous timestamp;

- **Kill_chain_phase:** phases of the kill chain[40] where the attack was detected.

Vulnerability: no specific properties for this SDO have been considered interesting for being included in the initial set, apart from the common ones

Attack Pattern: regarding this SDO, just one specific property has been considered interesting:

- **Kill_chain_phase:** phases of the kill chain where the attack was detected.

Tool: for how concern Tool SDO, the list of the interesting properties is composed by the following ones:

- **Label:** open-vocabulary field that indicates the kind of tool considered. Possible values could be "denial-of-service "vulnerability-scanning" and "remote-access", "privilege-escalation", "password-cracking", "password-sniffing", "memory-analysis", "reconnaissance;"
- **Kill_chain_phase:** phases of the kill chain where the attack that is using this tool was detected;
- **Tool_version:** version of the tool.

Threat Actor: interesting specific properties of Threat Actor SDO are the following:

- **Labels:** open-vocabulary field that indicates the type of the Threat Actor. Possible values could be "criminal", "hacker", "spy" and "terrorist;"
- **Aliases:** list of other names that this actor could use;
- **Roles:** open- vocabulary field that indicates a list of roles that the Threat Actor could play. Some examples could be "agent" and "director;"
- **Goals:** high level goals of the Threat Actor;
- **Sophistication:** open-vocabulary field which represents skills, training, expertise of the actor. Some examples could be "minimal", "intermediate", "advanced;"
- **Resource_level:** open-vocabulary field that represents the organizational level at which the actor works, which, in turn, determines the resource available for the attack. Some example could be "individual", "team", "government;"
- **Primary_motivation:** open-vocabulary field that represents the motivation of the Threat Actor. Some examples could be "accidental", "dominance", "personal-satisfaction", "revenge", "industrial-espionage", "sabotage", "hacktivism", "data-theft;"
- **Secondary_motivation:** same considerations as Primary_motivation;
- **Personal_motivation:** same considerations as Primary_motivation.

Identity: last SDO considered. The set of interesting specific properties is composed by the following ones:

[40] https://en.wikipedia.org/wiki/Kill_chain

- **Labels:** list of roles that this Identity performs (e.g., CEO, Domain Administrator, Doctor). No open-vocabulary defined for this property;
- **Identity_class:** open-vocabulary field that indicates the type of entity that this Identity describes. Some examples could be “individual”, “organization”, “unknown” and “group;”
- **Sectors:** open-vocabulary field, which represents the list of industry sectors that this Identity belongs to. Some examples could be “aerospace”, “automotive”, “defense” and “financial-services;”
- **Regions:** list of regions or geographic locations this Identity is located or operates in.

These features, when available from the IoCs coming from the monitored infrastructure, will be used in the training period of the heuristic analysis. As next steps, threshold definition and score computation will be performed by each of them and new features will be identified, in order to calibrate the engine and refine our tool, with the aim of improving the overall procedure used for assigning the threat score to security information coming from OSINT-data sources.

More precisely, other already existing SDOs properties could be considered, as well as, thanks to the extensibility of STIX standard, specific custom properties, not yet defined in the standard itself, that could be created ad-hoc, following STIX naming guidelines described in the official documentation [OAS17], for representing particular features that could improve our heuristic analysis, These tasks of training and engine calibration will be done in close collaboration with the partners involved in the DISIEM pilot deployments and the implementation of the OSINT data fusion and analysis tools. The results will be included in next deliverable D4.2 - OSINT data fusion and analysis architecture.

6 Summary and Conclusions

This deliverable presents an in-depth analysis of security-related OSINT sources, which can be classified as structured or unstructured. Additionally, the dark web is considered a special class due to its specific characteristics and requirements. These classes differ mostly in the type of content, in the format of the information and in the tools required to extract it. Depending on the type of source, information is collected by means of parsers and crawlers (possibly custom-built), by available APIs or by specialized commercial services.

A complete list of OSINT sources specified by the DiSIEM industrial partners has been compiled. Information from these sources is being collected, which forms the basis for various case-studies regarding the processing of OSINT to integrate relevant information into the SIEMs.

The literature review revealed that most work that uses OSINT in a security-related context is related to collecting infrastructure-specific information; to the collection and extraction methodologies; to the correlation of user behaviour with OSINT; to feed protection systems with OSINT; to the gathering of exploit data; and to black-listed IPS.

Regarding the processing and analysis of OSINT, there are some open-source general purpose tools that can be extended for that purpose. The alternatives are paid tools and security-related news feeds.

The ongoing work on the models and techniques to process OSINT is firstly described in this deliverable. This will be the main theme of the next deliverable in work package 4.

Finally, the deliverable provides the first insights on how the relevant OSINT data can be merged with infrastructure-related IoCs, and communicated and shared between software components and the SIEM.

References

[ALI16] AlienVault. (2016). AlienVault Open Threat Exchange (OTX) TM User Guide, 1–44.

[ALQ16] S. S. Alqahtani, E. E. Eghan, and J. Rilling. Tracing known security vulnerabilities in software repositories–a semantic web enabled modeling approach. *Science of Computer Programming*, 121:153–175, 2016.

[AND17] Andongabo, A. & Gashi, I. (2017). vepRisk - A Web Based Analysis Tool for Public Security Data. 13th European Dependable Computing Conference, 4-8 Sep 2017, Geneva, Switzerland.

[CAM13] Rodrigo Campiolo, Luiz Arthur F. Santos, Daniel Macêdo Batista, and Marco Aurélio Gerosa. 2013. Evaluating the Utilization of Twitter Messages As a Source of Security Alerts. In 28th Annual ACM Symposium on Applied Computing.

[CAP] "CAPEC™ Common Attack Pattern Enumeration and Classification," [Online]. Available: https://capec.mitre.org/.

[CEB10] Cebula, J. J., & Young, L. R. (2010). A Taxonomy of Operational Cyber Security Risks. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, (December), 1–47.

[CHA16] C.-Y. Chang, Z. Teng, and Y. Zhang. Expectation-regulated neural model for event mention extraction. In Proceedings of NAACL-HLT, pages 400–410, 2016.

[CON17] Fengyu Cong, Andrew Leung, and Qinglai Wei. *Advances in Neural Networks-ISNN*. Springer, 2017.

[COR16] A. Correia, P. Ferreira, and A. Bessani. Descoberta de ameaças de segurança através do twitter. In INForum, 2016.

[CYB] "Cyber Observable eXpression (CybOX™) Archive Website," [Online]. Available: https://cyboxproject.github.io/.

[DAN07] R. Danyliw, J. Meijer and Y. Demchenko, "The Incident Object Description Exchange Format," December 2007. [Online]. Available: https://www.ietf.org/rfc/rfc5070.txt.

[DIS] "DiSIEM D2.2 – Reference architecture and integration plan", 2017.

[DIS17] "DiSIEM D2.1 In-depth analysis of SIEMs extensibility", 2017.

[ENI17] Enisa. (2017). Incident Handling Automation. Retrieved June 29, 2017, from https://www.enisa.europa.eu/topics/csirt-cert-services/community-projects/incident-handling-automation.

[Ern06] Ernesto Damiani, Sabrina De Capitani di Vimercati, Pierangela Samarati, Marco Viviani, S. D. (2006). A WOWA-based Aggregation Technique on Trust Values Connected to Metadata. Electronic Notes in Theoretical Computer Science, 131-142.

[ESP16] A. d. M. Del Esposte, R. Campiolo, F. Kon, and D. Batista. A collaboration model to recommend network security alerts based on the mixed hybrid approach. In Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2016.

[EDK15] M. Edkrantz, S. Truvé, and A. Said. Predicting vulnerability exploits in the wild. In Cyber Security and Cloud Computing (CSCloud), 2015 IEEE 2nd International Conference on, pages 513–514. IEEE, 2015.

[ERK15] Y. Erkal, M. Sezgin, and S. Gunduz. A new cyber security alert system for twitter. In 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pages 766–770. IEEE, 2015.

[FAR13] SANS, "Tools and Standards for Cyber Threat Intelligence Projects," https://www.sans.org/reading-room/whitepapers/warfare/tools-standards-cyber-threat-intelligence-projects-34375, 2013.

[FIE14] R. Fielding and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Authentication," Internet Engineering Task Force (IETF), 2014. [Online]. Available: https://www.rfc-editor.org/rfc/pdfrfc/rfc7235.txt.pdf.

[FOR17] Forum of Incident Response and Security Teams.: Common Vulnerability Scoring System v3.0 Speci_cation Document, Technical Paper, available at: https://www.first.org/cvss/specification-document, last accessed on July 2017.

[FRA15] F. Fransen, A. Smulders and R. Kerkdijk, "Cyber security information exchange to gain insight into the effects of cyber threats and incidents," E & I Elektrotechnik und Informationstechnik 132 (2): 106–112, 2015.

[Jos12] Gil-Lafuente, José M. Merigó and Anna M. (2012). Decision-making techniques with similarity measures and OWA operators, Sort (Statistics and Operations Research Transactions), 36, 81-102

[JEN16] Jenhani, F., Gouider, M. S., & Said, L. Ben. (2016). A Hybrid Approach for Drug Abuse Events Extraction from Twitter. Procedia Computer Science, 96(September), 1032–1040. https://doi.org/10.1016/j.procs.2016.08.121

[JON15] C. L. Jones, R. A. Bridges, K. M. Huffer, and J. R. Goodall. Towards a relation extraction framework for cyber-security concepts. In Proceedings of the 10th Annual Cyber and Information Security Research Conference, page 11. ACM, 2015.

[KAL14] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[KAM14] P. Kampanakis, "Security Automation and Threat Information-Sharing Options," IEEE Security & Privacy, vol. 12, no. 5, pp. 42-51, 2014.

[KER15] D. Kergl. Enhancing network security by software vulnerability detection using social media analysis extended abstract. In Data Mining Workshop (ICDMW), 2015 IEEE International Conference on, pages 1532–1533. IEEE, 2015.

[KIM14] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[KÜH14] Kührer, M., Rossow, C., & Holz, T. (2014). Paint it black: Evaluating the effectiveness of malware blacklists.

[LES14] J. Leskovec, A. Rajaraman, and J. D. Ullman. Mining of massive datasets. Cambridge University Press, 2014.

[LIA16] X. Liao, K. Yuan, X. Wang, Z. Li, L. Xing, and R. Beyah. Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pages 755–766. ACM, 2016.

[LIB16] LIBERATORE, S. (2016). What happens in an internet second: 54,907 Google searches, 7,252 tweets, 125,406 YouTube video views and 2,501,018 emails sent. DailyMail, [online] p.1. Available at: http://www.dailymail.co.uk/sciencetech/article-3662925/What-happens-internet-second-54-907-Google-searches-7-252-tweets-125-406-YouTube-video-views-2-501-018-emails-sent.html [Accessed 4 Jul. 2017].

[LIU15] Y. Liu, A. Sarabi, J. Zhang, P. Naghizadeh, M. Karir, M. Bailey, and M. Liu. Cloudy with a chance of breach: Forecasting cyber security incidents. In 24th USENIX Security Symposium (USENIX Security 15), pages 1009–1024, 2015.

[MAR16] Marinos, L. (2016). ENISA threat taxonomy: A tool for structuring threat information. Initial report., (January), 1–24. Retrieved from https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/enisa-threat-landscape/etl2015/enisa-threat-taxonomy-a-tool-for-structuring-threat-information

[MAT12] M. L. Mathews, P. Halvorsen, A. Joshi, and T. Finin. A collaborative approach to situational awareness for cybersecurity. In Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference on, pages 216–222. IEEE, 2012.

[MCA] McAfee, “Data Exchange Layer”, [Online]. Available https://www.mcafee.com/us/solutions/data-exchange-layer.aspx.

[MCN13] N. McNeil, R. A. Bridges, M. D. Iannacone, B. Czejdo, N. Perez, and J. R. Goodall. Pace: Pattern accurate computationally efficient bootstrapping for timely discovery of cyber-security concepts. In Machine Learning and Applications (ICMLA), 2013 12th International Conference on, volume 2, pages 60–65. IEEE, 2013.

[MIL11] B. A. Miller, M. S. Beard, and N. T. Bliss. Eigenspace analysis for threat detection in social networks. In Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on, pages 1–7. IEEE, 2011.

[MIT] MITRE, “Common Vulnerability and Exposures. The Standard for Information Security Vulnerability Names”, MITRE Website, https://www.mitre.org/

[MIT16] S. Mittal, P. K. Das, V. Mulwad, A. Joshi, and T. Finin. Cybertwitter: Using twitter to generate alerts for cybersecurity threats and vulnerabilities. In International Symposium on Foundations of Open Source Intelligence and Security Informatics. IEEE Computer Society, 2016.

[MOO96] T. K. Moon. The expectation-maximization algorithm. IEEE Signal processing magazine, 13(6):47–60, 1996.

[MOR12] S. More, M. Matthews, A. Joshi, and T. Finin. A knowledge-based approach to intrusion detection modeling. In Security and Privacy Workshops (SPW), 2012 IEEE Symposium on, pages 75–81. IEEE, 2012.

[MUL11] V. Mulwad, W. Li, A. Joshi, T. Finin, and K. Viswanathan. Extracting information about security vulnerabilities from web text. In Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on, volume 3, pages 257–260. IEEE, 2011.

[NER09] F. Neri and P. Geraci. Mining textual data to boost information access in osint. In 2009 13th International Conference Information Visualisation, pages 427–432. IEEE, 2009.

[NUN16] E. Nunes, A. Diab, A. Gunn, E. Marin, V. Mishra, V. Paliath, J. Robertson, J. Shakarian, A. Thart, and P. Shakarian. Darknet and deepnet mining for proactive cybersecurity threat intelligence. In Intelligence and Security Informatics (ISI), 2016 IEEE Conference on, pages 7–12. IEEE, 2016.

[OAS17] OASIS, “STIX™ Version 2.0. Part 1: STIX Core Concepts,” 2017. [Online]. Available: https://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part1-stix-core.pdf.

[OAS171] OASIS, “STIX™ Version 2.0. Part 2: STIX Objects,” 2017. [Online]. Available: https://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part2-stix-objects.pdf.

[OAS172] OASIS, “STIX™ Version 2.0. Part 3: Cyber Observable Core Concepts,” 2017. [Online]. Available: https://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part3-cyber-observable-core.pdf.

[OAS173] OASIS, “STIX™ Version 2.0. Part 4: Cyber Observable Objects,” 2017. [Online]. Available: https://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part4-cyber-observable-objects.pdf.

[OAS174] OASIS, “STIX™ Version 2.0. Part 5: STIX Patterning,” 2017. [Online]. Available: https://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part5-stix-patterning.pdf.

[OAS175] OASIS, “TAXII™ Version 2.0,” 2017. [Online]. Available: https://docs.oasis-open.org/cti/taxii/v2.0/taxii-v2.0.pdf.

[OPE] “OpenIOC - An Open Framework for Sharing Threat Intelligence,” [Online]. Available: http://www.openioc.org/.

[RES15] R. J., “The 'Basic' HTTP Authentication Scheme,” 2015. [Online]. Available: https://www.rfc-editor.org/rfc/pdfrfc/rfc7617.txt.pdf.

[RIT15] A. Ritter, E. Wright, W. Casey, and T. Mitchell. Weakly supervised extraction of computer security events from twitter. In Proceedings of the 24th International Conference on World Wide Web, pages 896–905. ACM, 2015.

[ROS10] Rossow, C., Czerwinski, T., Dietrich, C. J., & Pohlmann, N. (2010). Detecting Gray in Black and White. MIT Spam Conference.

[SAB15] C. Sabottke, O. Suciu, and T. DumitraÈ™. Vulnerability disclosure in the age of social media: exploiting twitter for predicting real-world exploits. In 24th USENIX Security Symposium (USENIX Security 15), pages 1041–1056, 2015.

[SAN13] L. A. F. Santos, R. Campiolo, M. A. Gerosa, D. M. Batista, and C. Mourao-PR-Brasil. Detecção de alertas de segurança em redes de computadores usando redes sociais. In Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2013.

[SAU17] C. Sauerwein, C. Sillaber, A. Mussmann and R. Breu, "Threat Intelligence Sharing Platforms: An Exploratory Study of Software Vendors and Research Perspectives," 13th International Conference on Wirtschaftsinformatik (WI2017), S. 837-851, 2017.

[SHA15] Shackleford, D. (2015). Who’ s Using Cyberthreat Intelligence and How ? SANS.

[SIN08] Sinha, S., Bailey, M., & Jahanian, F. (2008). Shades of Grey: On the effectiveness of reputation based black-lists. Proceedings of the International Conference on Malicious and Unwanted Software(Malware), 57–64.

[STR] “Structured Threat Information eXpression (STIX™) 1.x Archive Website,” [Online]. Available: http://stixproject.github.io/.

[TRU1] “Trusted Automated eXchange of Indicator Information (TAXII™) 1.x Archive Website,” [Online]. Available: https://taxiiproject.github.io/.

[W2V] Tomas Mikolov and team. word2vec: Vector Representations of Words (Tensorflow). https://www.tensorflow.org/tutorials/word2vec. Accessed: 2017-07-17.

[WAN15] Peng Wang, Jiaming Xu, Bo Xu, Cheng-Lin Liu, Heng Zhang, Fangyuan Wang, and Hongwei Hao. Semantic clustering and convolutional neural network for short text categorization. In *ACL (2)*, pages 352–357, 2015.

[ZAK14] M. J. Zaki, W. Meira Jr, and W. Meira. Data mining and analysis: fundamental concepts and algorithms. Cambridge University Press, 2014.

[ZHA11] S. Zhang, D. Caragea, and X. Ou. An empirical study on using the national vulnerability database to predict software vulnerabilities. In International Conference on Database and Expert Systems Applications, pages 217–231. Springer, 2011.

[ZHU16] Z. Zhu and T. Dumitras. Featuresmith: Automatically engineering features for malware detection by mining the security literature. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pages 767–778. ACM, 2016.

List of Acronyms

Acronym	Description
ACDC	Advanced Cyber Defence Center
ANN	Artificial Neural Networks
CVE	Common Vulnerabilities and Exposures
CSIRTs	Computer Security Incident Response Teams
CVSS	Common Vulnerability Scoring System
CTI	Cyber Threat Intelligence
DDOS	Distributed Denial of Service
ENISA	European Agency for Network and Information Security
EM	Expectation-Maximization
GRU	Gated Recurrent Neural Networks
HUMINT	Human Intelligence
IODEF	Incident Object Description and Exchange Format
IoC	Indicators of Compromise
IPS	Intrusion Prevention Systems
IPA	Japanese Information-technology Promotion Agency
LDA	Latent Dirichlet Allocation
LSTM	Long Short Term Memory
NVD	National Vulnerability Database
ISAC council	National Council of Information Sharing and Analysis Center
NLP	Natural Language Processing
Open DXL	Open Data eXchange Layer
OSINT	Open Source Intelligence
SOC	Security Operation Center
SOCMINT	Social Media Intelligence
SaaS	Software as a Service
SDO	STIX Domain Objects
SRO	STIX Relationship Objects
STIX	Structured Threat Information eXpression
SVM	Support Vector Machines
TTPs	Tactics, Techniques and Procedures
TC	Technical Committee
TF	Term Frequency
TABI	Trust Assessment of Blacklists Interface
TAXII	Trusted Automated eXchange of Indicator Information
DHS	U.S. Department of Homeland Security

Appendix A – OSINT sources

In this Appendix is presented a list of the OSINT sources used by the various partners of the project, divide by categories. Table 7 presents the sources divided by category, while Section 0 presents Twitter accounts.

News sites	
Name	**Source**
Dark Reading	http://www.darkreading.com/
Computer World	http://www.computerworld.com/
European Union Agency for Network and Information Security	https://www.enisa.europa.eu/
Security Focus	http://www.securityfocus.com/headlines
Blogs	
Name	**Source**
Schneier on Security	https://www.schneier.com/
Dancho Danchev's Blog	http://ddanchev.blogspot.pt/
Network Security Blog	http://www.mckeay.net/
Risky.biz	https://risky.biz/
Kai Roer’s Security Culture Ramblings	https://roer.com/
IPs for whitelists	
Name	**Source**
awesome-threat-intelligence	https://github.com/hslatman/awesome-threat-intelligence
Cisco Umbrella - Umbrella Popularity List	http://s3-us-west-1.amazonaws.com/umbrella-static/index.html
IPs for blacklists	
Name	**Source**
AutoShun.org	https://www.autoshun.org/
ThreatMiner – Data Mining for Threat Intelligence	https://www.threatminer.org/
Spamhaus	https://www.spamhaus.org/
Suspicious Domains	https://isc.sans.edu/suspicious_domains.html
I-Blocklist	https://www.iblocklist.com/lists
badips_cyrusauth	https://www.badips.com/get/list/cyrusauth/age=1d
badips_squid	https://www.badips.com/get/list/squid/?age=1d
security_research	http://security-research.dyndns.org/pub/botnet/ponmocup/ponmocup-finder/ponmocup-infected-domains-latest.txt
alienvault	https://reputation.alienvault.com/reputation.data
lists_blocklist_ssh	https://lists.blocklist.de/lists/ssh.txt
badips_apache-overflows	https://www.badips.com/get/list/apache-overflows/?age=1d
ci-badguys	http://cinsscore.com/list/ci-badguys.txt

emergingthreats_compromised-ips	http://rules.emergingthreats.net/blockrules/compromised-ips.txt
lists_blocklist_ircbot	https://lists.blocklist.de/lists/ircbot.txt
badips_apache-dokuwiki	https://www.badips.com/get/list/apache-dokuwiki/?age=1d
badips_apache-defensible	https://www.badips.com/get/list/apache-defensible/?age=1d
badips_Php-url-fopen	https://www.badips.com/get/list/Php-url-fopen/?age=1d
nothink_http	http://www.nothink.org/blacklist/blacklist_malware_http.txt
badips_qmail-smtp	https://www.badips.com/get/list/qmail-smtp/?age=1d
badips_apache-scriddies	https://www.badips.com/get/list/apache-scriddies/?age=1d
badips_apache-noscript	https://www.badips.com/get/list/apache-noscript/?age=1d
badips_pop3	https://www.badips.com/get/list/pop3/?age=1d
badips_bruteforce	https://www.badips.com/get/list/bruteforce/?age=1d
nothink_irc	http://www.nothink.org/blacklist/blacklist_malware_irc.txt
badips_pureftpd	https://www.badips.com/get/list/pureftpd/?age=1d
virustotal	https://www.virustotal.com/vtapi/v2/ip-address/report
dshield	http://www.dshield.org/ipsascii.html?limit=10000
badips_local-exim	https://www.badips.com/get/list/local-exim/?age=1d
lists_blocklist_bots	https://lists.blocklist.de/lists/bots.txt
badips_proxy	https://www.badips.com/get/list/proxy/?age=1d
badips_php-cgi	https://www.badips.com/get/list/php-cgi/?age=1d
lists_blocklist_imap	https://lists.blocklist.de/lists/imap.txt
badips_drupal	https://www.badips.com/get/list/drupal/?age=1d
badips_nginx	https://www.badips.com/get/list/nginx/?age=1d
badips_dovecot-pop3	https://www.badips.com/get/list/dovecot-pop3/?age=1d
badips_sql	https://www.badips.com/get/list/sql/?age=1d
badips_unknown	https://www.badips.com/get/list/unknown/?age=1d
badips_proftpd	https://www.badips.com/get/list/proftpd/?age=1d
badips_sip	https://www.badips.com/get/list/sip/?age=1d
badips_imap	https://www.badips.com/get/list/imap/?age=1d
badips_http	https://www.badips.com/get/list/http?age=1d
malc0de	http://malc0de.com/bl/IP_Blacklist.txt
badips_ftp	https://www.badips.com/get/list/ftp/?age=1d
badips_assp	https://www.badips.com/get/list/assp/?age=1d
badips_vsftpd	https://www.badips.com/get/list/vsftpd/?age=1d
lists_blocklist_bruteforcelogin	https://lists.blocklist.de/lists/bruteforcelogin.txt
badips_apacheddos	https://www.badips.com/get/list/apacheddos/?age=1d
badips_xmlrpc	https://www.badips.com/get/list/xmlrpc/?age=1d
lists_blocklist_strongIP	https://lists.blocklist.de/lists/strongips.txt
badips_postfix	https://www.badips.com/get/list/postfix/?age=1d
badips_phpids	https://www.badips.com/get/list/phpids/?age=1d
badips_wp	https://www.badips.com/get/list/wp/?age=1d
lists_blocklist_ftp	https://lists.blocklist.de/lists/ftp.txt
badips_sql-attack	https://www.badips.com/get/list/sql-attack/?age=1d
nothink_ssh	http://www.nothink.org/blacklist/blacklist_ssh_day.txt
badips_pureftp	https://www.badips.com/get/list/pureftp/?age=1d

badips_courierauth	https://www.badips.com/get/list/courierauth/?age=1d
badips_plesk-postfix	https://www.badips.com/get/list/plesk-postfix/?age=1d
badips_vnc	https://www.badips.com/get/list/vnc/?age=1d
badips_dns	https://www.badips.com/get/list/dns/?age=1d
badips_exim	https://www.badips.com/get/list/exim/?age=1d
badips_ssh	https://www.badips.com/get/list/ssh/?age=1d
badips_wordpress	https://www.badips.com/get/list/wordpress/?age=1d
zeustracker	https://zeustracker.abuse.ch/blocklist.php?download=badips
badips_sasl	https://www.badips.com/get/list/sasl/?age=1d
badips_apache-spamtrap	https://www.badips.com/get/list/apache-spamtrap/?age=1d
badips_ssh-ddos	https://www.badips.com/get/list/ssh-ddos/?age=1d
badips_rdp	https://www.badips.com/get/list/rdp/?age=1d
dragonForce_VNCPROBE	https://dragonresearchgroup.org/insight/vncprobe.txt
urlvir	http://www.urlvir.com/export-ip-addresses/
badips_default	https://www.badips.com/get/list/default/?age=1d
dragonForce_SSH	https://dragonresearchgroup.org/insight/sshpwauth.txt
badips_ssh-blocklist	https://www.badips.com/get/list/ssh-blocklist/?age=1d
badips_apache-wordpress	https://www.badips.com/get/list/apache-wordpress/?age=1d
badips_nginxpost	https://www.badips.com/get/list/nginxpost/?age=1d
badips_apache	https://www.badips.com/get/list/apache/?age=1d
badips_apache-w00tw00t	https://www.badips.com/get/list/apache-w00tw00t/?age=1d
badips_nginxproxy	https://www.badips.com/get/list/nginxproxy/?age=1d
badips_sql-injection	https://www.badips.com/get/list/sql-injection/?age=1d
badips_cms	https://www.badips.com/get/list/cms/?age=1d
feodotracker	https://feodotracker.abuse.ch/blocklist/?download=ipblocklist
lists_blocklist_apache	https://lists.blocklist.de/lists/apache.txt
badips_w00t	https://www.badips.com/get/list/w00t/?age=1d
badips_sshd	https://www.badips.com/get/list/sshd/?age=1d
badips_ssh-auth	https://www.badips.com/get/list/ssh-auth/?age=1d
badips_courierpop3	https://www.badips.com/get/list/courierpop3/?age=1d
cryptophp_master	https://raw.githubusercontent.com/fox-it/cryptophp/master/ips.txt
badips_smtp	https://www.badips.com/get/list/smtp/?age=1d
badips_badbots	https://www.badips.com/get/list/badbots/?age=1d
badips_apache-nohome	https://www.badips.com/get/list/apache-nohome/?age=1d
danger_rulez	http://danger.rulez.sk/projects/bruteforceblocker/blist.php
lists_blocklist_mail	https://lists.blocklist.de/lists/mail.txt
emergingthreats_botcc	http://rules.emergingthreats.net/blockrules/emerging-botcc.rules
turris_greylist	https://www.turris.cz/greylist-data/greylist-latest.csv
badips_owncloud	https://www.badips.com/get/list/owncloud/?age=1d
openbl	https://www.openbl.org/lists/base_30days.txt
badips_username-notfound	https://www.badips.com/get/list/username-notfound/?age=1d
IPList_IPset	https://raw.githubusercontent.com/firehol/blocklist-

	ipsets/master/firehol_level1.netset
badips_screensharingd	https://www.badips.com/get/list/screensharingd/?age=1d
malwaredomainlist	http://www.malwaredomainlist.com/updatescsv.php
dragonForce_HTTP	https://dragonresearchgroup.org/insight/http-report.txt
ransomwaretracker	https://ransomwaretracker.abuse.ch/feeds/csv
badips_spam	https://www.badips.com/get/list/spam/?age=1d
labs_snort	http://labs.snort.org/feeds/ip-filter.blf
badips_sshddos	https://www.badips.com/get/list/sshddos/?age=1d
badips_ddos	https://www.badips.com/get/list/ddos/?age=1d
cert	http://www.cert.org/downloads/mxlist.ips.txt
cruzit	http://www.cruzit.com/xwbl2csv.php
badips_apache-phpmyadmin	https://www.badips.com/get/list/apache-phpmyadmin/?age=1d
badips_postfix-sasl	https://www.badips.com/get/list/postfix-sasl/?age=1d
lists_blocklist_sip	https://lists.blocklist.de/lists/sip.txt
badips_telnet	https://www.badips.com/get/list/telnet/?age=1d
badips_dovecot-pop3imap	https://www.badips.com/get/list/dovecot-pop3imap/?age=1d
badips_apache-php-url-fopen	https://www.badips.com/get/list/apache-php-url-fopen/?age=1d
badips_apache-404	https://www.badips.com/get/list/apache-404/?age=1d
badips_dovecot	https://www.badips.com/get/list/dovecot/?age=1d
badips_asterisk	https://www.badips.com/get/list/asterisk/?age=1d
badips_apache-modsec	https://www.badips.com/get/list/apache-modsec/?age=1d
badips_named	https://www.badips.com/get/list/named/?age=1d
badips_asterisk-sec	https://www.badips.com/get/list/asterisk-sec/?age=1d
osint	http://osint.bambenekconsulting.com/feeds/c2-ipmasterlist-high.txt
autoshun	https://www.autoshun.org/download/?api_key=
badips_rfi-attack	https://www.badips.com/get/list/rfi-attack/?age=1d
badips_spamdyke	https://www.badips.com/get/list/spamdyke/?age=1d
sslbl	https://sslbl.abuse.ch/blacklist/sslipblacklist.csv
charles	http://charles.the-haleys.org/ssh_dico_attack_hdeny_format.php/hostsdeny.txt
BinaryDefense	https://www.binarydefense.com/banlist.txt
Talos	http://www.talosintelligence.com/feeds/ip-filter.blf
Domains/Botnets	
Name	**Source**
MalwareINT	https://intel.malwaretech.com/
Bambenek Consulting	http://osint.bambenekconsulting.com/feeds/c2-ipmasterlist.txt
Ransomware Tracker	http://ransomwaretracker.abuse.ch
Zeus Tracker	https://zeustracker.abuse.ch/
DNS-BH – Malware Domain Blocklist by RiskAnalytics	http://www.malwaredomains.com/
Snort/Suricata	
Name	**Source**

Proofpoint emerging threats intelligence	http://rules.emergingthreats.net/blockrules/
Hail a TAXII	http://hailataxii.com/
Bro	
Name	**Source**
CriticalStack intel feed	https://intel.criticalstack.com/
Firewall rules	
Name	**Source**
Proofpoint emerging threats intelligence	http://rules.emergingthreats.net/fwrules/
OWASP Core Rule Set	https://github.com/SpiderLabs/owasp-modsecurity-crs
Malware	
Name	**Source**
OPSWAT Metadefender	https://www.metadefender.com/threat-intelligence-feeds
VirusShare	https://virusshare.com/
MISP - Open Source Threat Intelligence Platform & Open Standards For Threat Information Sharing	http://www.misp-project.org/features.html
IP reputation	
Name	**Source**
CINSscore	http://cinsscore.com/list/ci-badguys.txt
Yara rules	
Name	**Source**
Yara-rules	https://github.com/Yara-Rules/rules
DNS sinkholes	
Name	**Source**
Bambenek Consulting	http://osint.bambenekconsulting.com/feeds/c2-dommasterlist-high.txt
IP address sinkholes	
Name	**Source**
Bambenek Consulting	http://osint.bambenekconsulting.com/feeds/c2-ipmasterlist-high.txt
Bad Domains	
Name	**Source**
Zeustracker	https://zeustracker.abuse.ch/blocklist.php?download=baddomains
Ransomware	
Name	**Source**
Ransomware Tracker	https://ransomwaretracker.abuse.ch/downloads/RW_IPBL.txt
Ransomware Tracker	https://ransomwaretracker.abuse.ch/downloads/LY_C2_DOMBL.txt
Ransomware Tracker	https://ransomwaretracker.abuse.ch/downloads/CW_C2_URLBL.txt
Phising sites	
Name	**Source**

OpenPhish	https://openphish.com/feed.txt
TOR nodes IPs	
Name	**Source**
dan.me.uk	https://www.dan.me.uk/torlist/
TOR project	https://check.torproject.org/exit-addresses
Various	
Name	**Source**
RiskIQ	https://www.riskiq.com/ https://www.riskiq.com/blog/ https://www.riskiq.com/products/security-intelligence-services/
Shodan	https://www.shodan.io/about/products
Blocklist.de	https://lists.blocklist.de/lists/all.txt
Computer Incident Response Center	https://www.circl.lu/doc/misp/feed-osint/
botvrij.eu	http://www.botvrij.eu/data/feed-osint/
inThreat	https://feeds.inthreat.com/osint/misp/
Pastebin	https://pastebin.com/

Table 7 - The OSINT sources used by the partners of the project, divided by category.