SAFECARE

Integrated cyber-physical security for health services

Specification of the IT threat detection system

Deliverable 5.1

Lead Author: CCS

Contributors: ISEP, CSI, ENC, PEN

Deliverable classification: PU

Co-funded by the Horizon 2020 Framework Programme of the European Union

Version Control Sheet

Title	Specification of the IT threat detection system
Prepared By	CCS
Approved By	ENC and PEN
Version Number	1.0
Contact	samantha.dauguetdemailly@airbus.com

Revision History:

Version	Date	Summary of Changes	Initials	Changes Marked
1.0	12/11/2019	First complete version	TO, MG, SD, DL	
1.1	19/11/2019	Modifications after ENC's and PEN's reviews	SD	

The research leading to these results has received funding from the European Union's Horizon 2020 Research and Innovation Programme, under Grant Agreement no 787002.

Contents

List of Figures

List of Tables

List of Acronyms

API	Application Programming Interface
APT	Advanced Persistent Threat
BMS	Building Management System
CE	Community Edition
CIDR	Classless Inter Domain Routing
DICOM	Digital Imaging and COMmunications in medicine
DNS	Domain Name System
DoA	Description of Action (also referred to as Description of Work)
ECHR	European Convention on Human Rights
EHR	Electronic Health Records
EU	European Union
FHIR	Fast Healthcare Interoperability Resources
FTP	File Transfer Protocol
FN	False Negative
FP	False Positive
GDPR	General Data Protection Regulation [4]
GNU GPLv3	GNU General Public License version 3
GPU	Graphics Processing Unit
HDFS	Hadoop Distributed File System
HTTP	HyperText Transfer Protocol
IAM	Identity and Access Management
ICMP	Internet Control Message Protocol
ID	Identifier
IDS	Intrusion Detection System
IE	Innovation Element
IG	Information Gain
Infs	Infinities
IP	Internet Protocol address
IPS	Intrusion Prevention System
IT	Information Technology
KNN	K Nearest Neighbours
ML	Machine Learning
NaN	Not a Number
NFS	Network File System
NIPS	Network Intrusion Prevention System
NSM	Network Security Monitoring
OISF	Open Information Security Foundation
OSI	Open Systems Interconnection
PACS	Picture Archiving and Communication System
PCAP	(Network) packet capture
PDF	Portable Document Format
PE	Portable Executable
RFE	Recursive Feature Elimination
ROC	Receiver Operating Curve
SFS	Sequential Feature Selection

SIEM	Security Information and Event Management
SMB	Server Message Block
SMTP	Simple Mail Transfer Protocol
SOAR	Security Orchestration, Automation and Response
SOC	Security Operation Center
SSL	Secure Sockets Layer
TAP	Terminal Access Point
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TP	True Positive
UDP	User Datagram Protocol
USB	Universal Serial Bus
WP	Work Package
YAML	YAML Ain't Markup Language

The SAFECARE Project

Over the last decade, the European Union has faced numerous threats that quickly increased in their magnitude, changing the lives, the habits and the fears of hundreds of millions of citizens. The sources of these threats have been heterogeneous, as well as weapons to impact the population. As Europeans, we know now that we must increase our awareness against these attacks that can strike the places we rely upon the most and destabilize our institutions remotely. Today, the lines between physical and cyber worlds are increasingly blurred. Nearly everything is connected to the Internet and if not, physical intrusion might rub out the barriers. Threats cannot be analysed solely as physical or cyber, and therefore it is critical to develop an integrated approach in order to fight against such combination of threats. Health services are at the same time among the most critical infrastructures and the most vulnerable ones. They are widely relying on information systems to optimize organization and costs, whereas ethics and privacy constraints severely restrict security controls and thus increase vulnerability. The aim of this proposal is to provide solutions that will improve physical and cyber security in a seamless and cost-effective way. It will promote new technologies and novel approaches to enhance threat prevention, threat detection, incident response and mitigation of impacts. The project will also participate in increasing the compliance between security tools and European regulations about ethics and privacy for health services. Finally, project pilots will take place in the hospitals of Marseille, Turin and Amsterdam, involving security and health practitioners, in order to simulate attack scenarios in near-real conditions. These pilot sites will serve as reference examples to disseminate the results and find customers across Europe.

Executive Summary

The objective of SAFECARE is to bring together the most advanced technologies from the physical and cyber security spheres to achieve a global optimum for systemic security and for the management of combined cyber and physical threats and incidents, their interconnections and potential cascading effects. The project focuses on health service infrastructures and works towards the creation of a global protection system, which covers threat prevention, threat detection, threat response and, in case of failure, mitigation of impacts across infrastructures, populations and environment.

One of the main outcomes of the SAFECARE project is a cyber threat detection system that aims at improving the detection of APTs and zero-day attacks on IT and BMS systems. The IT threat detection system is a part of the broader cyber threat detection system, as shown in Figure 1. The IT threat detection system supports the objective of improving incident detection by providing network monitoring and producing relevant information to improve the detection of zero-day attacks.

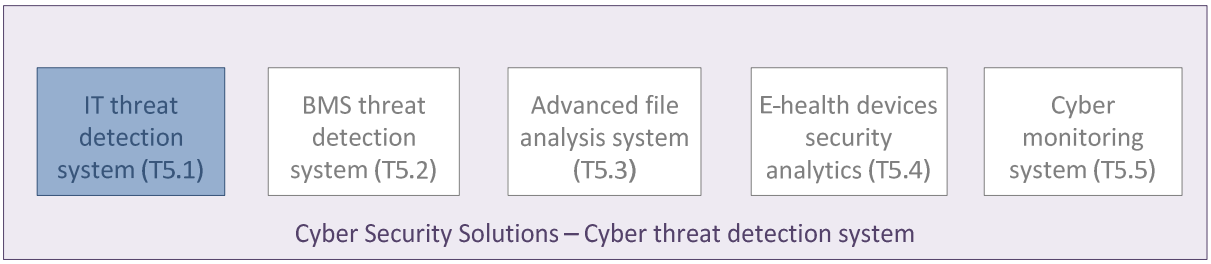

Figure 1 – IT threat detection system context

The specification of the IT threat detection system lists the functional requirements that the solution must meet. The requirements are taken from the Description of Work of Task 5.1 [1] and the requirements analysis of Deliverable 3.4. It also includes security requirements regarding ethics and data privacy.

The current solution is based on the network threat detection engine Suricata and is associated with a correlation engine Graylog. Other tools are also embedded to meet the requirements. The solution needs to be interconnected with the advanced file analysis system and the cyber threat monitoring system. The advanced file analysis system performs an in-depth analysis of the files extracted from network traffic by the IT threat detection system, and the cyber threat monitoring systems receives all security events produced by the IT threat detection system.

This document provides the specification of machine learning functionalities to be developed in the frame of the SAFECARE project in order to detect APTs and zero-day attacks on IT systems.

1 Introduction

As systems grow, encompassing more and more computer and devices — whose variety also increases — their attack surface skyrockets. At the same time, the stake is increasingly attractive to attackers whose methods improve every day.

An IT threat detection system must be able to detect known malware and attacks as well as suspicious behaviour that may be the work of a new unknown method for an attacker to slip into or harm the system, and therefore endanger the organisation and human lives it may protect. Such a frightening scenario already happened; in November 2019 for instance, a ransomware infected Rouen's CHU (university hospital); health professionals were forced to shut down the entire IT system and it had a drastic impact on the hospital's functioning — as of today, patient files are still not recovered. To reach this goal, a hybrid approach was chosen, combing both non-supervised and supervised methods. Non-supervised methods are based on the network threat detection engine Suricata which enables fast analysis based on malware/threat signature.

As illustrated in Figure2, the IT threat detection system is part of the cyber threat detection systems of the SAFECARE project and belongs to the cyber security solutions (Work Package 5). Like all cyber threat detection systems, the IT threat detection system is connected to the cyber threat monitoring system.

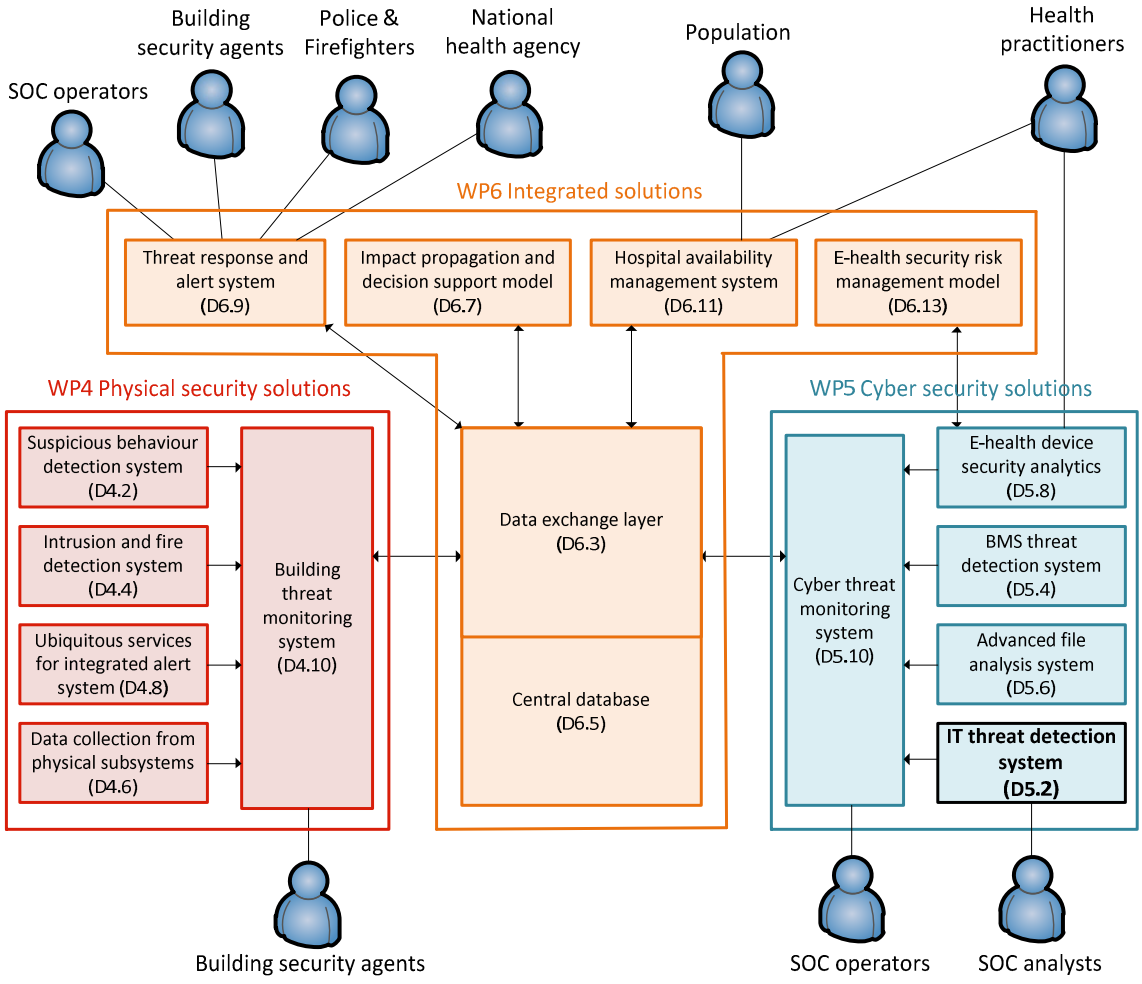

Figure 2 – SAFECARE global architecture

The requirements are based on the Description of Work of the Grant Agreement and the requirements analysis of Deliverables 3.4 and 3.9. Requirements are presented in Section 2.

Interconnections with other systems of the SAFECARE project, such as the advanced file analysis system and the cyber threat monitoring system, are described in Section 3.

The solution is presented in Section 4 including a network threat detection engine, a web interface, a packet capture solution and a correlation engine. It is also introduced the simulation platform and the integration architecture.

Section 5 specifies the new functionalities to be implemented, that is the machine learning architecture and machine learning algorithms.

Finally a mapping table is provided in Section 6 to meet the requirements listed in Section 2 with the functionalities of the solution.

2 Functional requirements

This section describes the requirements for the IT threat detection system, as provided by the Grant Agreement [1], SAFECARE Deliverables 3.4 [2] and 3.9 [3].

Guiding document	Resulting requirements categories
Grant Agreement	DoA requirements
SAFECARE Deliverable 3.4	End-user requirements
SAFECARE Deliverable 3.9	Ethics and privacy requirements

Table 1 – Guiding documents and resulting requirements categories

2.1 DoA requirements

The following requirements are based on the description of action/work of SAFECARE's Grant Agreement regarding the IT threat detection system (Task 5.1) as well as on the description of innovation element number 5 (IE5: An IT oriented threat detection system and analytics tools to improve cyber threat investigation), which is directly linked to Task 5.1.

The main objective is to improve network traffic incident/threat detection and investigation. In order to do so, the IT threat detection system will use a hybrid approach combining both non-supervised methods and supervised methods.

2.1.1 Non-supervised methods

The IT threat detection system must monitor network traffic in order to detect cyber threats by using attack and vulnerability pattern matching;

This non supervised approach must be improved to enhance cyber threat detection and to limit the number of false positive and false negative events.

New detection capabilities, completing the usual pattern matching methods, must be embedded. Those new features will rely on metadata extracted from the network traffic and enable identifying abnormal behaviour on the network.

2.1.2 Supervised methods

The IT threat detection system must improve the detection of zero-day attacks — which go undetected by usual pattern matching methods — in the target environment. Resulting requirements are that the IT threat detection system must implement:

- A supervised approach based on machine learning;
- Smart visualisation interfaces in order to enhance the apprehension of new cyber-attacks.

Additional requirements apply to the supervised approach based on machine learning:

- Algorithms must be robust, scalable and deployable anywhere without performance loss;
- Relevant indicators must be identified to discard parameters dependent on the site (i.e. the environment).

2.1.3 Interconnections

Overall, security events (incidents, alerts) resulting from network traffic monitoring and zero-day attacks detection system must be communicated to the cyber threat monitoring system (D5.10) which is the IT threat detection system's interface with the central database (D6.3) through the data exchange layer (D6.3), as shown in Figure 3.

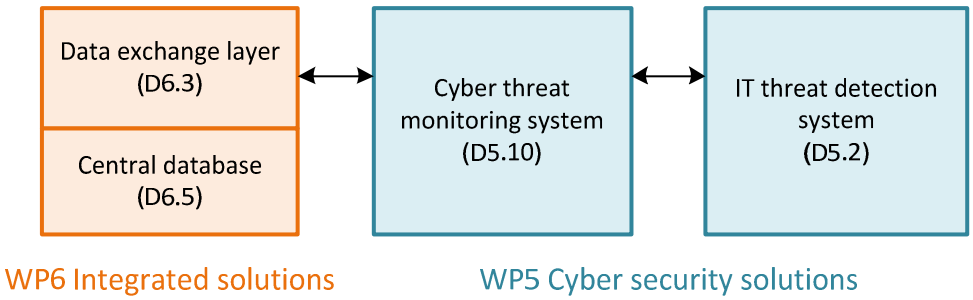

Figure 3 – Communications from the IT threat detection system to the central database

2.1.4 Training requirements

The final implementation of the IT threat detection system requires a training phase during which:

- Attack scenarios will be performed on a simulated environment representing the information system of the target infrastructure in order to build gold datasets;
  Gold datasets are understood to be a good representative sample of data, consisting of a set of labelled network traffic packets. Labels are for instance crisis flags, incident IDs or attack patterns.
- Supervised algorithms must be trained on the test platform and then on each demonstration site, relying on a real traffic.
  Their training should enable them to automatically classify network traffic data when suspicious behaviour is encountered and therefore simplify threat pattern investigation by SOC analysts.

2.2 End-user requirements

The end-users requirements described in this section are a result of the requirements analysis made for Deliverable 3.4 during Task 3.3 of the SAFECARE project.

SAFECARE's end-users being health and security practitioners, the main objective of Task 3.3 was to determine their requirements in terms of physical security solutions, cyber security solutions, crisis management, communication and coordination strategies. Functional requirements were identified, in particular concerning threat detection. Their adaptation to the case of the IT threat detection system as a subsystem is as follows.

2.2.1 Specialization and protocol-awareness

One major requirement is that the IT threat detection system must be specialized to the healthcare environment, taking into account healthcare-specific threats and healthcare-specific protocols such as protocols used by the X-Ray intervention system: the FHIR protocol, to communicate with the Epic Electronic Health Records (EHR) system, and the DICOM protocol, to communicate with a Picture Archiving and Communication System (PACS).

2.2.2 Integrated approach

Another essential requirement is that the whole solution must have an integrated approach: events from the IT threat detection system must be correlated with threats from other SAFECARE detection systems, such as threats from the physical world. This is taken into account by requirements from Section 2.1.3.

2.2.3 Performances
Another major requirement is that the IT threat detection system must enable fast and early detection of suspicious events.

2.2.4 Threat detection capabilities
Regarding abnormal events to take into account, intrusions must be detected and identified.

2.2.5 Keeping up-to-date
The IT threat detection system must enable its update with new patterns for important (yet) unpatched vulnerabilities. The updating process must verify the updates' authenticity and integrity through digital signature controls.

2.2.6 User interface
An additional requirement is that the IT threat detection system must implement an ergonomic user interface. Access to this environment must be secured through authentication and authorization control.

2.2.7 Testability
Additionally, the IT threat detection system must include functionalities to enable and facilitate independent testing. A security test should not be incorrectly classified as an actual attack.

2.2.8 Interoperability
Lastly, the IT threat detection system must fit in with the existing security system.

2.3 Ethics and privacy requirements
The ethics and privacy requirements described in this section are a result of the analysis made for Deliverable 3.9 during Task 3.6 of the SAFECARE project regarding ethics, privacy, and confidentiality constraints. Deliverable 3.9's analysis takes into account Europeans regulations that are the General Data Protection Regulation (GDPR)[4] and the European Convention on Human Rights (ECHR)[5] as well as state specific laws.

2.3.1 Ethics constraints
Four main moral principles have been identified when dealing with ethical decision-making in a health environment:

- Principle of respect for the persons and autonomy;
- Principle of justice;
- Principle of non-maleficence;
- Principle of beneficence.

Additionally, three secondary principles were also identified:

- Responsibility;
- Dignity;
- Accountability.

Those seven principles are described in details in Deliverable 3.9. They must be considered as constraints and be taken into account when outlining and implementing the IT threat detection system.

2.3.2 Privacy constraints

2.3.2.1 The right to privacy

The European Convention on Human Rights article 8 protects the right to respect for private and family life and therefore the protection of health data. It stipulates that a State must not interfere with the enjoyment and exercise of individual's fundamental rights and if it does, even due to an omission or a lack of action, it will be held responsible. There are however specific cases in which interference with the right to respect privacy may happen but it must be: in accordance with the law, pursue one or more of the exhaustively cited legitimate aims, and necessary in a democratic society.

In the SAFECARE context, it means that State authorities have to ensure that privacy guarantees are complied with healthcare services.

2.3.2.2 The right to data protection

The right to data protection is closely linked to the right to privacy but remains different. Indeed, the right to privacy refers to the right to respect for private and family life and is thus limited to the private sphere of a person, whereas the right to data protection is relevant whenever data is being processed.

The GDPR states principles that must be applied to the processing of personal data, namely:

- the principle of purpose limitation;
  The data must be "collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes"[6]. They are, however, exceptions to this principle, such as archiving purposes in the public interest, scientific research, historical research or statistical purposes.
- the data minimisation principle;
  The data processing must be limited to what is strictly needed to achieve a legitimate goal. To reduce the possibility to make a connection between data and its subject, solutions such as anonymization or pseudonymisation should be implemented
- the storage limitation principle;
  The storage time of personal data must only equal the time needed to complete the purposes for which data is processed.
- the integrity and confidentiality principle;
  Data must be "processed in a manner that ensures appropriate security of the personal data, including protection against unauthorised or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organisational measures" [7].
- the lawfulness, fairness and transparency principle;
  Concerning lawfulness, processing of personal data is allowed on one or more of the following grounds: consent, performance of a contract, compliance with a legal obligation, protection of vital interests of the data subject or someone else, public interest and the legitimate interests of the data controller (i.e. any official body determining the purpose and means of the processing of personal data) or a third party.
- the principle of accuracy;
  Personal data must be accurate and up-to-date.
- the principle of accountability;
  The data controller must be able to show that it complies with the GDPR principles.

2.4 List of requirements

The following table gathers requirements identified in the previous sections.

Identifier	Requirement
	Non-supervised methods
R-NSUP-01	The IT threat detection system must monitor network traffic in order to detect cyber threats by using attack and vulnerability pattern matching. This is the usual non-supervised method provided by IDS systems
R-NSUP-02	The IT threat detection system's non-supervised methods must be improved to enhance cyber threat detection and to limit the number of false positive and false negative events
R-NSUP-03	The IT threat detection system's non-supervised methods must embed new capabilities relying on metadata extracted from the network traffic and enabling identification of abnormal behaviour on the network
	Supervised methods
R-SUP-01	The IT threat detection system must implement a supervised approach based on machine learning
R-SUP-02	The supervised algorithms must be robust, scalable and deployable anywhere without performance loss
R-SUP-03	Relevant indicators must be identified to discard site adherent parameters in the supervised algorithms
R-SUP-04	The IT threat detection system must implement smart visualisation interfaces in order to enhance the apprehension of new cyber-attacks
	Specialization and protocol-awareness
R-SPE-01	The IT threat detection system must be specialized to the healthcare environment, taking into account healthcare-specific threats and healthcare-specific protocols
	General detection functionalities
R-FUNC-01	The IT threat detection system must be able to detect and identify intrusions
	Transversal functionalities
R-FUNC-02	The IT threat detection system must enable its update with new patterns for important (yet) unpatched vulnerabilities. The updating process must verify the updates' authenticity and integrity through digital signature controls.
R-FUNC-03	The IT threat detection system must implement an ergonomic user interface. Access to this environment must be secured through authentication and authorization control.
	Interconnections
R-CON-01	Overall, security events (incidents, alerts) resulting from network traffic monitoring and zero-day attacks detection system must be communicated to the cyber threat monitoring system
	Training and testing requirements
R-TRA-01	During the training phase of the IT threat detection system, attack scenarios must be performed on a simulated environment representing the information system of the target infrastructure in order to build gold datasets
R-TRA-02	During the training phase of the IT threat detection system, supervised algorithms must be trained on the test platform
R-TRA-03	During the training phase of the IT threat detection system, supervised algorithms must be trained on each demonstration site, relying on a real traffic
R-TRA-04	The supervised algorithms must be trained to automatically classify network traffic data when suspicious behaviour is encountered
R-TRA-05	The IT threat detection system must include functionalities to enable and facilitate independent testing
	Performances and interoperability

R-PER-01	The IT threat detection system must enable fast and early detection of suspicious events
R-PER-02	The IT threat detection system must fit in with the existing security system
	Ethics and privacy
R-REG-01	The IT threat detection system must comply with the privacy and data protection requirements applicable under EU law and particularly with the provisions of the General Data Protection Regulation.

Table 2 – List of requirements

3 Interconnections

The objective of the IT threat detection system is to detect zero-day attacks in critical health infrastructure. To this aim, it monitors the network traffic and analyses IT events sent by the IT infrastructure that is supervised. IT events are generally provided as software logs.

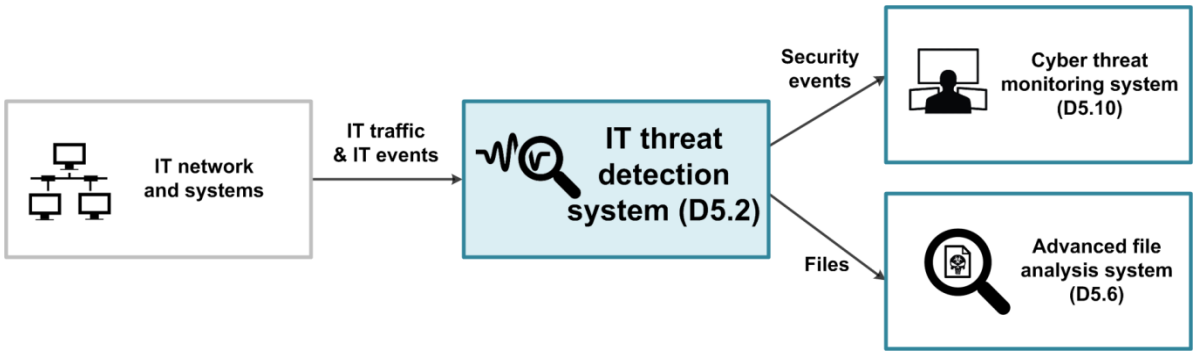

Figure 4 – Interconnections of the IT threat detection system

As illustrated in Figure 4, the IT threat detection system generates security events that are received by the cyber threat monitoring system. The IT threat detection system also extracts files from the IT network traffic in order to submit the files to the advanced file analysis system.

3.1 Interconnection with the critical health infrastructure

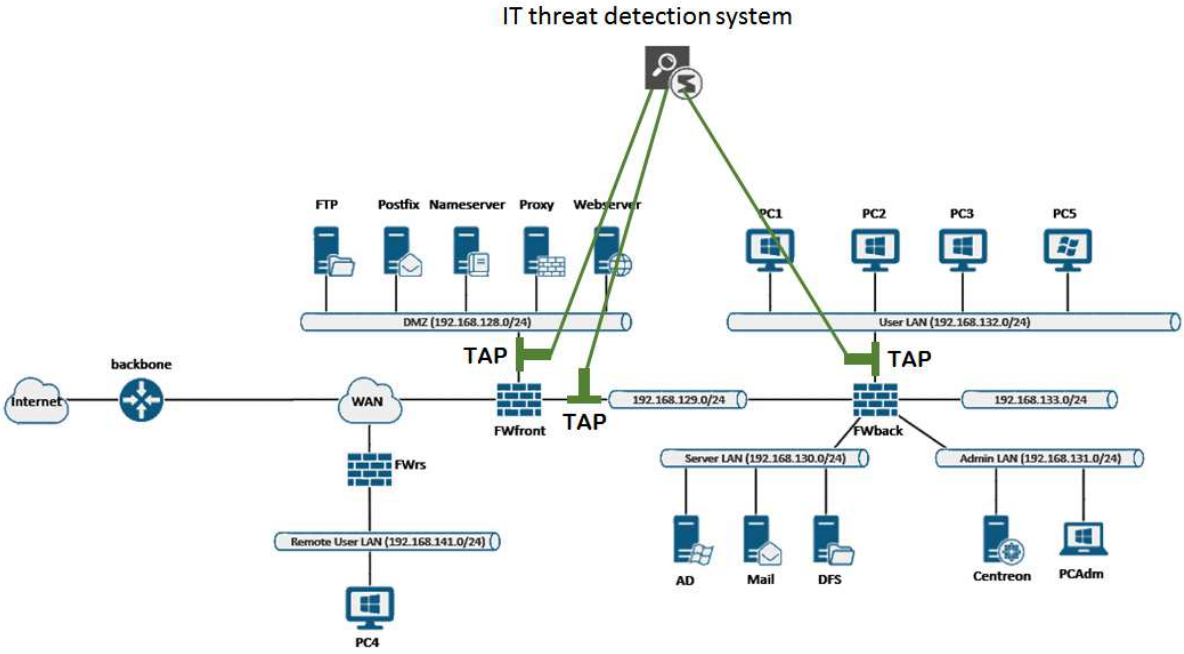

Figure 5 – Generic deployement of the IT threat detection system

The critical health infrastructure, which will be supervised, is composed of an IT infrastructure that the IT threat detection system will monitor. The IT infrastructure can be broken down into two categories: network and host systems.

The network traffic, which transits thanks to network components such as switches or routers, is analyzed by the IT threat detection system. The network traffic is generally duplicated so that the IT threat detection system can inspect the network traffic in real time without being too intrusive. The traffic duplication can be carried out by using a network TAP (Terminal Access Point) or by

port mirroring of a network switch. The IT threat detection system can detect network attacks using signatures that correspond to attack patterns.

IT events (such as USB stick connections) from host systems (servers, virtual machines, operating systems …) are expected to be sent to the IT threat detection system in order to detect intrusions. Sending IT events must be configured on the supervised IT infrastructure.

Each final deployment of the IT threat detection system will be more specific to target IT infrastructures and will be specified as soon as precise information about those infrastructures is provided.

3.2 Interconnection with the cyber threat monitoring system

The cyber threat monitoring system centralizes security events from cyber threat detection systems including the IT threat detection system. The cyber threat monitoring system provides visualization interfaces so that SOC operators can react to attacks and prevent or mitigate damage by viewing the assets impacted by attacks.

Security events are produced by the IT threat detection system and then sent to the cyber threat monitoring system. A security event is generated when matching a network signature or triggering a security rule implemented within the IT threat detection system. Security rules can be based on statistics, aggregations and correlations regarding IT events.

SOC operators can call back and view the IT threat detection system from the cyber threat monitoring system to analyze why security events have been triggered in order to confirm an alert as a cyber security incident.

3.3 Interconnection with the advanced file analysis system

The advanced file analysis system detects malwares in critical health infrastructure by performing an in-depth analysis of files. However, the advanced file analysis system requires the submission of a file by a person or system. For operators, manually submitting large numbers of files to the advanced file analysis system is a laborious and difficult task without errors. That's why the IT threat detection system will automatically extract files from network traffic and send them to the advanced file analysis system.

The IT threat detection system has a file extraction capability and will include a connector for automatically submitting files to the advanced file analysis system. After the file analysis has been performed by the advanced file analysis system, the IT threat detection system gets the analysis result and sends a security event including network metadata to the cyber threat monitoring system in case of malware detection.

4 System design

The IT threat detection system includes a network threat detection engine as well as a web interface, a packet capture solution and a correlation engine.

4.1 Network threat detection engine

The network threat detection engine is based on the open source software Suricata. Suricata is a detection engine that captures the network, detects attacks and creates alerts or journals logs.

The Suricata engine is capable of real time intrusion detection (IDS), inline (i.e. in the direct communication path between source and destination) intrusion prevention (IPS), network security monitoring (NSM) and offline processing of network traffic captures (better known as packet captures (pcap)).

The Suricata project and code is owned and supported by the Open Information Security Foundation (OISF), a non-profit foundation organized to build a next generation IDS/IPS engine.

In the context of SAFECARE and the IT threat detection system, Suricata is used as an IDS, to monitor network traffic in order to detect cyber threats. Suricata comes with existing rulesets to which are added rules and signatures designed for the SAFECARE project as a way to match healthcare-specific threats and anomalies. The rule framework is flexible and enables the administrator to turn threat intelligence and behavioural indicators into detection signatures; an alert will be raided when a match is found on the network.

Performance-wise, Suricata's network threat detection engine has been tested and is able to inspect traffics of several gigabits. This is made possible by the fact that the engine is multithreaded.

Protocols supported are TCP, UDP, ICMP and "IP" (which stands for "any") for transport layer protocols, as well as HTTP, FTP, TLS (including SSL), SMB and DNS (from v2.0) for application layer protocols. Any port (through which traffic comes and goes) can be inspected.

One important part for Suricata is to rebuild the traffic through the layers of the OSI model. To correctly achieve this task, Suricata must consider the IP Fragmentation, the TCP segmentation, and all invalidity type like wrong size, checksum, invalid TCP window size. For some protocols Suricata realizes decomposition and normalization.

First Suricata realizes and validates the IP defragmentation. After that the reconstruction is done on the TCP layer and the data flow is reconstituted. If the application exists in Suricata, the normalization is done through the data flow.

The entire configuration in Suricata is done in YAML (*YAML Ain't Markup Language*) files.

4.1.1 Rules Format

Suricata allows writing signature, a signature consists of an action, the header and the rules options. There are four types of actions; the action determines what happens when the signature matches. The header defines the protocol, IP addresses, ports and direction of the rule. The option rule defines the specifics of the rule.

Rule example:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"Example");
```

In this example, red is the action, blue is the header and green are the options.

4.1.2 Actions

The four types of actions are Pass, Drop, Reject and Alert. For the SAFECARE project, only two types of actions "Pass" and "Alert" will be used.

- Pass: if a signature matches, Suricata stops scanning the packet and skips to the end of all rules for the current packet.
- Alert: if a signature matches and contains "alert", the packet will be treated like any other non-threatening packet, except for this one an alert will be generated by Suricata. Only the system administrator can notice this alert.

Rules will be loaded in the order of which they appear in files. But they will be processed in a different order. Signatures have different priorities. The most important signatures will be scanned first. There is a possibility to change the order of priority. The default order is: pass, drop, reject, alert.

4.1.3 Header

The protocol is in first position in the header.

Rule example (tcp is the protocol):

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"Example");
```

Different keywords exist on different OSI layers. There are four basic protocols:

- tcp (for tcp-traffic)
- udp
- icmp
- ip (ip stands for 'all' or 'any')

There are also a few so-called application layer protocols, or layer 7 protocols that can be picked from (for example: "http", "ftp", "ssh", "smtp"…). The availability of these protocols depends on whether the protocol is enabled in the configuration file "suricata.yaml".

The next fields in the header are "Source" and "Destination".

Rule example:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"Example");
```

With source and destination, it is possible to specify the source of the traffic and the destination of the traffic, this can be an IP address or IP ranges. It is also possible to use variable and specify specific IP addresses in the configuration file.

The source and destination are followed by ports:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:”Example”);
```

The direction can also be specified, the direction tells in which way the signature has to match:

source -> destination (right directions)

source <> destination (both directions)

4.1.4 Rule options

The rule options, which are separated by semicolons, are found in parenthesis. Some options have settings others are just a keyword.

<keyword>: <settings>;
<keyword>;

The order in the option rule is important and its modification can change the meaning of the rules.

The file extraction works on top of the protocol parsers. The supported protocols are HTTP, SMTP, FTP, NFS and SMB.

This simplest rule will extract all file to disk:

```
alert http any any -> any any (msg:"FILE store all"; filestore; sid:1; rev:1;)
```

It is also possible to extract files:

- from a specific extension;
- or using their magic number (or “filemagic”), a constant numerical or text value used to identify a file format;
- or using checksums (md5, sha1 or sha256).

Within the scope of SAFECARE, specific rules will be implemented to extract DICOM files (Digital imaging and communications in medicine). Vulnerability was disclosed in the DICOM Part 10 file format. It is possible to include a Portable Executable (PE) malware within the preamble of a DICOM file. After being extracted from network traffic, the DICOM file will be sent to the advanced file analysis system.

4.2 Web interface

Scirius is a web interface for the ruleset management, as illustrated in Figure6. Scirius handles the rules file and update associated files.

Scirius CE is developed by Stamus Networks and is available under the GNU GPLv3 license.

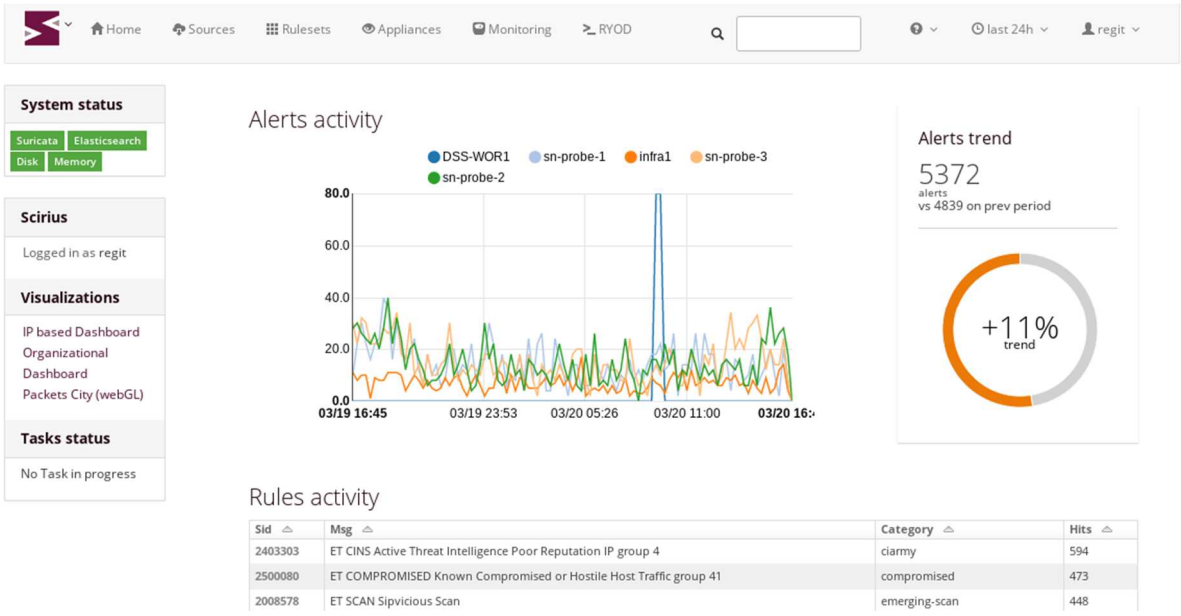

Figure 6 – Scirius screenshot example

4.3 Packet capture

Moloch is a full packet capture tool (illustrated in Figure7) which allows storing and indexing traffic in standard PCAP format. It's an open source project with a web application that provides PCAP browsing, searching, analysis and PCAP carving for exporting.

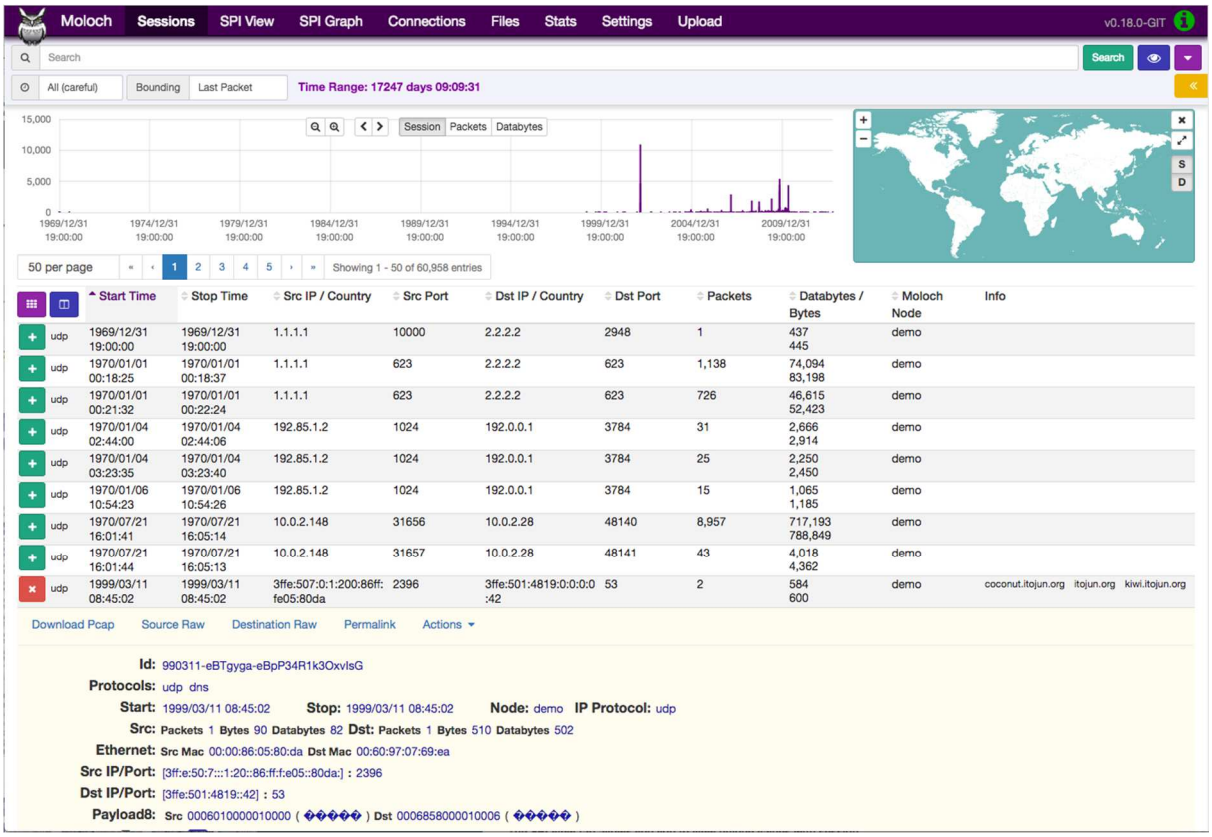

Figure 7 – Moloch screenshot example

4.4 Correlation engine

Supervising a system implies to deploy multiple technologies such as the ones previously described to detect, investigate, qualify, mitigate and prevent the manifestation of security situations. All these security components deployed over the network, all these agents installed into critical systems produce a lot of security events that need to be continuously assessed.

This could not be possible without the deployment of another technology: the SIEM. Indeed, one objective of this kind of software is to collect and centralize all the information produce by all the security components deployed on the system to supervise. This allows the cyber security experts to focus only on the SIEM instead of checking one by one each security component which would be not feasible.

The second advantage of a SIEM is to correlate the information to identify security problems. In fact, if it's possible to detect a situation that needs to be investigated from one source of information, most of the time, it is needed to correlate the information from multiple sources to identify more complex scenarios. A SIEM allows a user to define a set of rules that will trigger an alarm every time the situation described by a rule occurs. This alarm will be forwarded to a SOAR. This tool is used by the security experts to supervise a system. It can receive alerts from multiple sources that need to be qualified by the experts or, if needed, classified as false positive thus enabling them to be quantified.

The following figure explains the log collection from multiple inputs into the SIEM and the alert process to the SOAR.

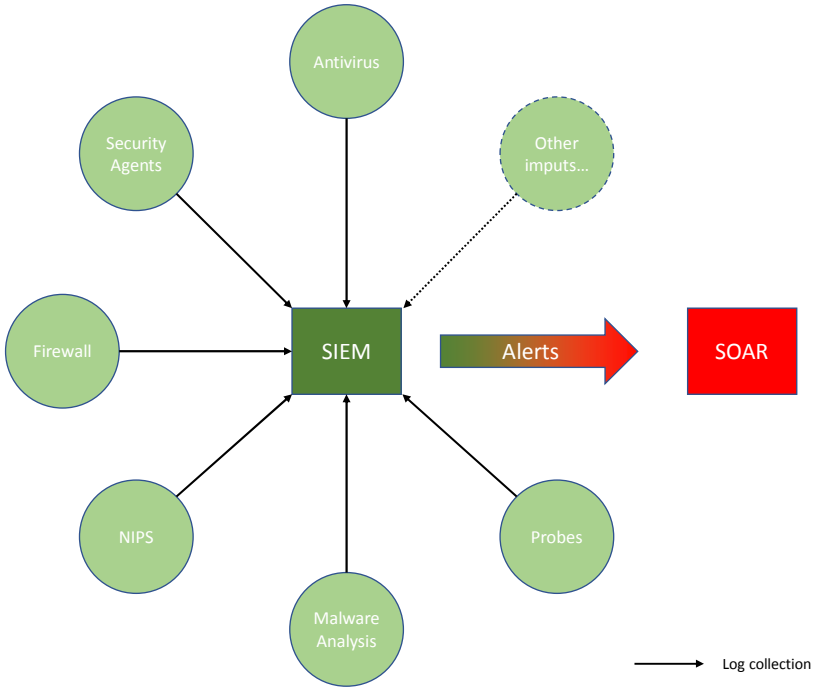

Figure 8 – SIEM integration

In the frame of the SAFECARE project, the SIEM Graylog will alert the SOAR Cymerius.

Graylog is an open source log management platform that can be used as a SIEM. It offers specific security-oriented features which can be easily extended by plugins.

In addition, dashboards can display the results of search query in Graylog. Graylog dashboards are visualizations or summaries of information contained in log events.

Cymerius is a SOAR solution developed by Airbus CyberSecurity SAS. This solution has been successfully deployed in multiple infrastructures in various contexts. By modelling the system to supervised, Cymerius can evaluate the impact of a security problem on the company's functional infrastructure, thus allowing security experts to focus on the most impacting problems. Indeed, when an alert is received, Cymerius can determine which assets and services are impacted. Cymerius will be used as the cyber threat monitoring system in the scope of SAFECARE and is presented extensively in Deliverable D5.9.

4.5 Simulation platform

Airbus CyberRange is a platform developed by Airbus CyberSecurity SAS which can be used for multiple purposes. Composed of servers and switches, it can be used to emulate a whole or a part of a system. In the frame of the SAFECARE project, it allows to host the whole supervision and investigation architecture.

The CyberRange can handle multiple workzones. A workzone is independent from the other ones and access rights are managed on each one. It's accessible from the internet, allowing partners to deploy their own virtual machines (or docker images) and network architecture themselves. This is a very powerful tool that will oil the wheels of the collaboration in the consortium for a fast architecture deployment.

The other advantage of the use of virtual machines is the possibility to upscale any cluster easily and fine-tune the resources allocated to each one. Indeed, it is possible to copy and deploy again any machine or docker image needed.

Moreover, a set of machines and their network links, which is called a topology, can be saved to be easily deployed. In addition, every machine deployed is backed up regularly to prevent any data loss.

4.6 Integration architecture

The integration architecture of the correlation engine and machine learning tools is presented in Figure 9.

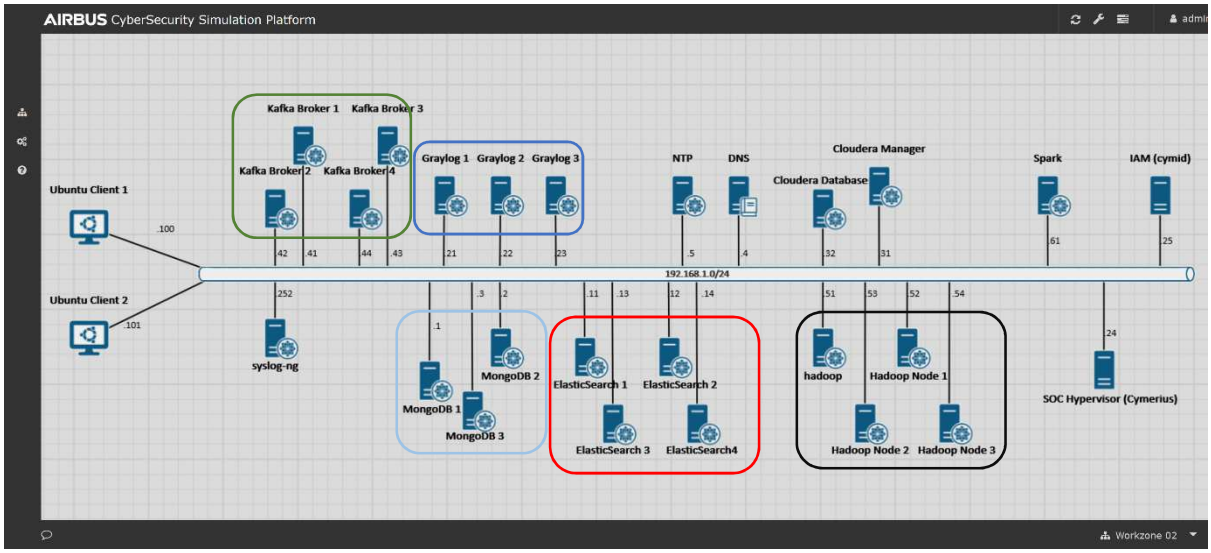

Figure 9 – CyberRange integration architecture

Here are some explanations from the main clusters:

- In blue, the Graylog cluster
- In light blue, the Mongo cluster, used by Graylog to store its configuration
- In red, the Elastic cluster, where all events are stored
- In green, the Kafka cluster. Kafka is used as a buffer. If there is too much event to handle for the Graylog cluster, Kafka will store these events on a queue, preventing from any data loss.
- In black, the Hadoop cluster, used for machine learning.

Some of the clusters have been deploy and are managed by Cloudera. On the far right, it is possible to see Cymerius and CymID which is an IAM solution.

5 Specification of new functionalities

5.1 Machine Learning Module

Threat detection systems emerged with the need to create systems capable of identifying and dealing with external and internal threats. They detect intrusions through the analysis of traffic information and logs and can be classified as misuse or signature-based and anomaly-based. Signature-based detection focuses on the analysis of network traffic, trying to identify sequences of bytes and packets that could be associated with threats and attacks. Therefore, the creation of signatures is dependent on previous knowledge of the type of traffic (benign and non-benign) expected in a given network[8]. This process has, however, two major limitations:

> (1) they are unable to detect zero-day or novelty attacks whose signature differs substantially from those of known attacks, i.e. previously unknown threats, and
>
> (2) threats capable of changing and evolving as is the case with metamorphic malware (malware capable of reprogramming itself when propagated) and polymorphic malware (malware that uses encryption to avoid detection and capable of self-mutation) are also hard to identify[9].

In the case of anomaly-based detection, there is a creation of a normal activity profile, that can later be used to cross-reference against eventual network or host activity to find anomalies, which can vary from unusual to malicious behaviour. The most prominent advantage of anomaly-based detection in comparison to misuse-based is its conduciveness to detect novelty attacks. Despite these benefits, anomaly-based detection is prone to false positives (mistaking normal traffic for an attack), costing operator time and money, more so than misuse detection. Lastly, it's still important to mention that false negatives (mistaking attacks for normal traffic) are a problem with both approaches and need to be carefully looked at when developing an IDS since it can put in danger the whole infrastructure[10].

Machine Learning methods have been widely used to detect threats since they can understand the behaviour of an attack using known traffic datasets, and then can detect attacks in the network. Mainly two types of machine learning approaches are being used: supervised learning and unsupervised learning. The former is often used in misuse-based detection since, because of the labelled data, it is possible to learn the signatures of certain known attacks. The latter is commonly used in anomaly-based detection since unsupervised algorithms usually create a normality profile and identify traffic patterns that deviate from that profile, effectively detecting anomalies without needing labels.

In SafeCare project we will use both approaches to build a ML engine that will help in the detection of new attack patterns and new vulnerabilities, and will be included in the IT threat detection system. The structure of this ML engine is described in Figure 10. The ML engine receives the network traffic and uses anomaly detection (unsupervised learning) and misuse-based (supervised learning models) models to detect attacks and anomalies. Then this information is outputted to a SIEM to help in the detection of security incidents.

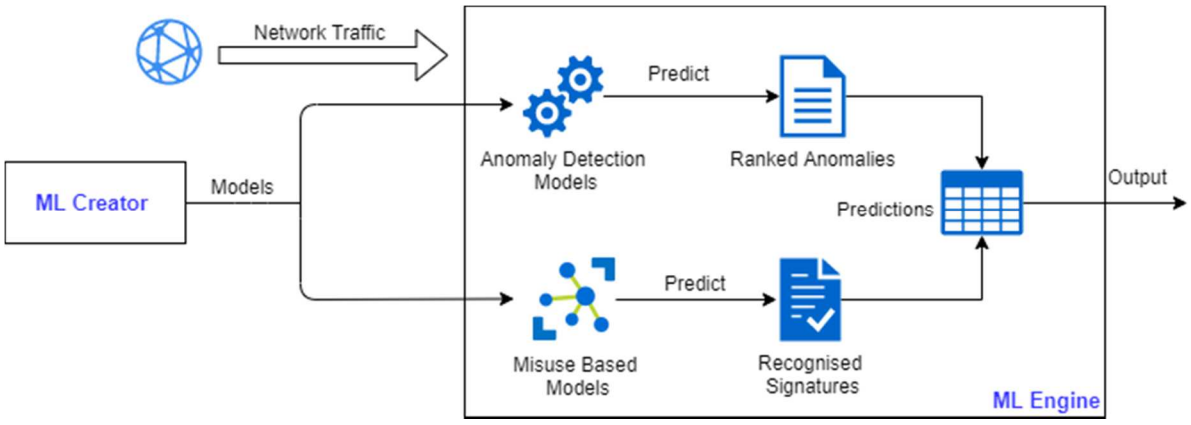

Figure 10 – ML Engine

In order to design robust and scalable machine learning algorithms that could be deployed on any site without loss of performance it is important to use varied data sets, relevant features and real data, from updated types of attacks to real hardware and software scenarios. Moreover, this data needs to be constantly updated due to the fast emergence of new attacks. Thus, new models need to be continually created to improve the performance of ML Engine. This task is performed by the ML creator described in more detail in the next section.

In the context of Safecare project, the continuous improvement of ML engine will be implemented in three incremental phases. In the first phase we will use the public available datasets that are intended to resemble real data traffic, such as ISCX2012 [11], and CICIDS2017 and CSE-CIC-IDS2018 [12], to create the first models to be included in the ML engine. We will use these datasets since they have several different attacks, typically used in the real networks, that allow the machine learning algorithms to learn how to detect them. In the second phase, the ML engine will be upgraded with data from the simulations of the Safecare's infrastructure on CyberRange. Then, the ML engine will be fine-tuned for each end-user scenario, in the demonstration phase.

All the models used on ML engine will be evaluated not only by the typical evaluation metrics related to attacks detection (see Section 5.2.4), but also considering its performance. To ensure the best performance we will investigate models deployment in different technologies, such as:

Apache Hadoop is a free and open source framework that makes easy massive distributed data processing. It provides a distributed and resilient storage (HDFS) and processing of big data with the MapReduce programming model. Hadoop facilitates the processing of a huge amount of data.

Apache Spark is also a free and open source cluster-computing framework that needs to be interfaced with a distributed storage system. It can be interfaced with a wide variety of storage systems including HDFS provided by Hadoop. The role of Apache Spark is to compute and process data in a scalable and fault tolerant way.

Elasticsearch, based on the Lucene library, is one of the most popular search engines. It is also free and open source, but it is not competing with Hadoop as Hadoop is made for processing huge amount of data whereas Elasticsearch is made for speed. Elasticsearch can also be used with Apache Spark.

Anaconda is a free and open source distribution of the Python programming language and helps for package management and deployment for scientific computing.

TensorFlow is an API developed by the Google Brain Team under the Apache License. This API has evolved to become a full library for machine learning applications such as neural networks. TensorFlow will be deployed and managed by Anaconda.

Scikit-learn is another machine learning library for the Python programming language that features various algorithms. It will be also deployed and managed by Anaconda.

5.2 Machine learning algorithms

Machine learning algorithms have been used with success to detect threats in network traffic. However, sometimes single models are not enough to ensure good accuracy results. Thus, a good solution is to independently train multiple models to solve the same problem and then combine them to get better results. This paradigm is named Ensemble Learning, and it has been widely used to detect intrusions and attacks. In the context of the Safecare project, we will implement several models that have been successfully used for intrusion detection, and deploy them on the ML engine. We will consider supervised, unsupervised and ensemble techniques.

Considering supervised learning algorithms, Decision Trees are a very simple and widely used algorithm, since it offers a very good trade-off between performance and interpretability, favouring the latter [13][14]. Decision Trees are also very used as a basic method for developing ensemble techniques. Random Forest[15] and Adaboost[16] are two examples of ensemble of Decision Trees very used in threat detection. While Random Forests are parallel in their computation of the trees (Bagging), Adaboost is sequential (Boosting) because each base model is primarily trained with the instances that the previous model has misclassified. LightGBM[17] is another boosting ensemble algorithm used with success in intrusion detection.

Unsupervised algorithms will also be included in the ML Engine since they are very promising in detecting APTs and zero-day attacks. Examples of unsupervised algorithms are Isolation Forest, Support Vector Machines[18], One Class Nearest Neighbours[19] and Auto-Encoders[20]. In the context of the Safecare project, we intend not only to use the existing algorithms (some of them already mentioned), but also create new ensemble methods by combining supervised, unsupervised and ensemble algorithms.

As described in the previous section, the ML engine includes several models (anomaly detection and misuse-based models) to execute its function. These models are created using the ML creator described in Figure 11. The creation process consists of three main stages: pre-processing, train and evaluation. All models developed in the Safecare project will follow this creation flow. Each main part will be described in the next sections.

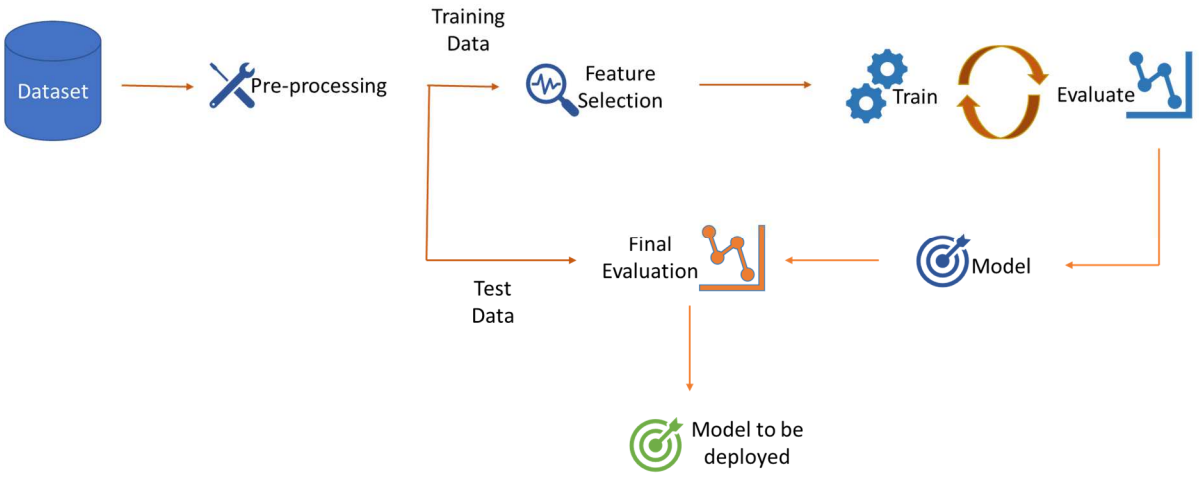

Figure 11 – Machine learning Creator (ML Creator)

5.2.1 Pre-Processing

Most raw traffic network information is not prepared to be fed into machine learning algorithms. As such most algorithms require some processing of the data in order to function properly. The most basic types of pre-processing include removal of invalid values such as NaN or Infs, removal of 0 variance features, feature scaling where the predictors are converted to the same scale, representation transformation where features are encoded into more appropriate data types and oversampling of the minority class and undersampling of the majority class, used to reduce class imbalance.

All the supervised learning algorithms mentioned above do not require any type of scaling but can benefit from the removal of invalid values, the removal of 0 variance predictors and representation transformation. As for the unsupervised methods with exception of isolation forest all the mention methods require some type of feature scaling.

A different kind of processing is feature selection and feature engineering. These techniques aim to reduce the number of variables of the data to remove noise and accelerate the models, and to create new features that have better relationships with the target variables. Additionally, these methodologies are centered around statistical methods and are often accompanied by the expertise of business professionals.

5.2.2 Feature Selection

Most of the machine learning algorithms can be trained using bidirectional network flows. Examples of significant features of each flow are:

- Duration of the flow
- Source and Destination IP
- Source and destination port
- Protocol
- Flow id
- Total number of packets in the forward (source to destination) and backward (destination to source) direction
- Size of each packet in both directions

- Number of packets per second in both direction
- Number of bytes per second
- Inter-arrival time of all packets in both directions
- Number of times every flag was set in both directions if applicable (FIN,SYN,RST,PSH,ACK,URG,CWR,ECE)
- Length of all headers in both directions
- Download and upload ratio
- Bytes sent in the initial windows of both direction

Due to the high number of features in each flow, feature selection is a very common process through which the relevance of certain features is evaluated regarding their correlation with the outputs. There are various techniques that can be used for this effect which can be divided into three categories:

- Wrapper methods, consisting of the optimization of the performance metric through the sub-setting of the features, examples of this are techniques such as Recursive Feature Elimination (RFE) and Sequential Feature Selection (SFS);
- Filter methods that use univariate statistics to measure intrinsic properties of the features and to rank them independently so that an engineer can manually subset them. Information Gain (IG), chi-square test and correlation coefficients are examples of this methods;
- Embedded methods, which look to add penalization to the objective function, making this penalization an integral part of the training of models. One example of this is L1 regularization [21].

Most of these methods are available when using supervised learning algorithms such as the previously described Random Forests and ensemble algorithms, which can use information gain and identify the best predictors due to their underlying tree-based models. However, when it comes to unsupervised models these methods are often not available since there are no labels that can be used to relate the features. As such techniques like dimensionality reduction are used in order to project the feature space into a new sub-space with fewer dimensions.

Moreover, some of the features can also be grouped to further increase the number of features for example if we got the size all the packets in a flow it is possible to create new features such as: “mean packet size”, “min packet size”, “max packet size”, “std packet size”. These features can increase the amount of information that our models have and if they turn out to not be relevant, they will just be discarded during the feature selection process.

5.2.3 Train and Evaluation

The training process happens after carefully splitting the available data into train and test set and putting away the testing set for later use (see Figure 11). If the data is enough the train set is further split into a validation dataset which is used to evaluate every model trained in the training set until a performance goal is achieved. In a last instance the model is tested on the test set and if the results are satisfactory the model is deployed. In the case of data shortage in the training set, the model can be trained using cross validation, where the training set is split in k parts or folds and the model is tested independently in each of the parts while being trained on the aggregation of the other k-1 parts. The performance is then calculated as the average of the performance of all folds.

In the deployment of any machine learning algorithm two important concepts are the robustness and the scalability. Regarding the scalability, it is important to note that machine learning models in general can suffer from concept drift[22] which means that their performance deteriorates over time when presented with a dynamic environment. In the case of cybersecurity this can mean that a model's detection rate can decreases if there are changes to the network traffic characteristics. There are various approaches to solve this problem. Some of which include the training of a model on a weekly or daily basis or in some extreme cases the use of online learning. For this effect the access to labelled data that is up to date is of great importance. Other common approaches include transfer learning where it is possible to use models trained in other datasets and map them to our specific use case. Perhaps more commonly, an analysist is used as a way of obtaining labelled data that is in theory up to date (active learning).

Previously described tree-based algorithms can be scaled by parallelizing their computations which can be done using libraries and frameworks such as Spark. Boosting models like LightGBM and XGBoost can even be trained using a GPU.

5.2.4 Performance Metrics

There are several metrics that can be used to evaluate the model before the deployment. Examples of these metrics are Accuracy, Receiver Operating Curve (ROC), Recall, Precision, and Specificity. These metrics are not always applicable, especially in the case of threat detection, where we are facing problems with high class imbalance and sometimes of multiclass nature. Accuracy is a very common metric, and it aims to show the proportion of correctly classified samples in the full scope of the classification:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

This metric is usually a good indicator of classifier performance. However, in the case of imbalanced data, as in the intrusion detection datasets, this measure is flawed since it favours the majority class (Benign traffic). Thus, this means that if the dataset was 90% benign traffic and the classifier failed to detect every attack but detected all normal traffic, it would still have an accuracy of 90%, which would only serve to mask the true performance of the model. Precision is the ratio between the number of correct positives (TP) and the total number of positives (TP+FP), i.e. $Precision = \frac{TP}{TP+FP}$.

In the case of intrusion detection, it would be the ratio between the number of correctly predicted attacks and the total amount of instances predicted as attacks. Precision is usually better than accuracy because it is not biased towards the majority class. Recall is the ratio of correctly predicted positive samples (TP) to the total number of positive samples (TP+FN), i.e. $Recall = \frac{TP}{TP+FN}$. In intrusion detection, it would be the ratio between the number of correctly predicted attacks and the total number of attacks. Sometimes it is necessary to use a single metric where both recall and precision can be incorporated. F1-score is this metric. It can be described as the harmonic mean between precision and recall, i.e., $F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}$. The F1-score is a good measure when it is desired that both metrics to have similar values. However, in some cases, when there is the need to optimize one over the other, it is often not the best metric to optimize.

5.3 Connector with the advanced file analysis system

As stated in section 3.3, the IT threat detection system will be interconnected with the advanced file analysis system (IE7), which is developed for Task 5.3 and specified in Deliverable D5.5. In order for such a connection to be possible, a specific connector will be developed. Its role is to enable the automatic submission of files extracted from the network traffic by the IT threat detection system to the advanced file analysis system.

The IT threat detection system, and more precisely the network threat detection engine, is configured so that each time a file can be and is extracted, the connector, embedded in the IT threat detection system, is called and given the file and its metadata. The connector then transmits the file to the advanced file analysis system. Once the analysis is done, the advanced file analysis system returns its result to the connector. Finally, the connector sends it, alongside file metadata, to the cyber threat monitoring system in the form of a log message.

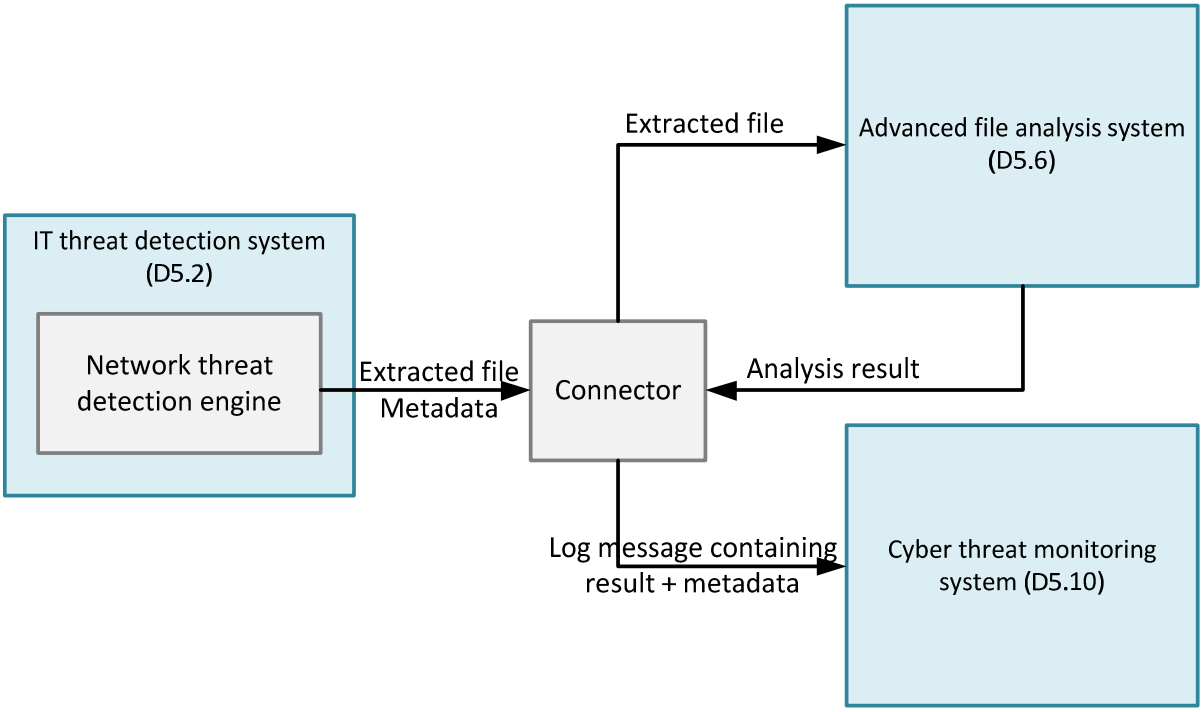

Figure 12 – Connector with the advance file analysis system

The connector, its technical implementation as well as ethics and privacy considerations regarding extracted files are specified thoroughly in Deliverable D5.5.

6 Requirements mapping

In Section 2, a list of functional requirements has been established. The following table maps the functional requirements to the existing and future functionalities of the solution.

Identifier	*Functionality*
R-NSUP-01	As described in Section 3.1, the solution will monitor network traffic using a network TAP or port mirroring a network switch. The solution uses a network intrusion detection system called Suricata, whose patterns can be configured, as described in Section 4.1.
R-NSUP-02	The combination of Suricata (NIDS), described in Section 4.1, with Graylog (used as a SIEM), described in Section 4.4, will allow enhancing cyber threat detection and reducing the number of false positive. Rules will be implemented in Graylog to correlate Suricata network alerts with IT events of the supervised IT infrastructure. The correlation feature will highlight low-level signal attacks and minimize recurrent false positive attacks. In addition, SOC experts will be able to classify and quantify false-positive and take them into account to fine tune Suricata and Graylog when possible.
R-NSUP-03	As described in Section 3.3, the solution will integrate a connector for submitting files to the advanced file analysis system. This connector will allow transmitting analysis results with metadata extracted from the network traffic to the cyber threat monitoring system. The abnormal behaviour on the network will be detected thanks to machine learning methods, as described in Section 5.1.
R-SUP-01	The solution will embed machine learning tools, as described in Section 5.1, to be used with machine learning algorithms described in Section 5.2.
R-SUP-02	The training process described in Section 5.2.3 will allow the algorithms to be robust, scalable and deployable anywhere without performance loss.
R-SUP-03	The pre-processing, described in Section 5.2.1, and the feature selection process, described in Section 5.2.2, will allow discarding site adherent parameters.
R-SUP-04	The smart visualisation interfaces will be implemented in Graylog with dashboards, as described in Section 4.4. When SOC operators receive alerts in the cyber threat monitoring system, they will be able to callback the Graylog tool to analyse alert logs and view dashboards related to alerts to apprehend new cyber-attacks.
R-SPE-01	Suricata can be configured with rules specifying protocols. As described in Section 4.1, the solution will be configured to extract DICOM files to submit these healthcare-specific files to the advanced file analysis system.
R-FUNC-01	The solution can detect network intrusions thanks to Suricata (NIDS) and host intrusions with Graylog receiving IT events as described in Section 3.1.

R-FUNC-02	New patterns can be added as set of rules, either in the network threat detection engine described in Section 4.1, or in the correlation engine described in Section 4.4. The authenticity and integrity of the updates of the solution will be verified using digital signatures.
R-FUNC-03	The ergonomic user interface is the web interface described in Section 4.2. Access to this environment will be secured through an identity and access management tool called CymID and mentioned in Section 4.6.
R-CON-01	As described in Section 3.2, all security events will be sent to the cyber threat monitoring system.
R-TRA-01	Attack scenarios can be performed on the simulation tool, described in Section 4.5, representing the information system of the target infrastructure.
R-TRA-02	The test platform will embed the simulation tool, described in Section 4.5. The training process, described in Section 5.2.3, will be performed during the training phase.
R-TRA-03	The training process, described in Section 5.2.3, will be performed during the training phase on each demonstration site, if possible relying on a real traffic. The real traffic assumption depends on having an end-user authorization during the demonstration phase.
R-TRA-04	The feature selection process, described in Section 5.2.2, will automatically classify network traffic data when suspicious behaviour is encountered.
R-TRA-05	The simulation tool, described in Section 4.5, will be used to emulate the IT threat detection system, thus facilitating independent testing.
R-PER-01	As described in Section 4.1, the Suricata engine is capable of real time intrusion detection. The correlation engine, described in Section 4.4, can generate security events in near real time.
R-PER-02	The correlation engine, described in Section 4.4, can receive events from almost any existing security system, as standards are used to transmit events.
R-REG-01	The solution will comply with the principle of purpose limitation as the data will be collected only for cyber threat detection. The solution will comply with the principle of data minimisation since only relevant data for cyber threat detection will be processed. The solution will comply with the principle of storage limitation as the data will be stored only the time needed for cyber threat detection. The solution will comply with the principle of integrity and confidentiality since an identity and access management tool, called CymID and mentioned in Section 4.6, will be used.

Table 3 – Mapping between requirements and functionalities

7 Conclusion

In the frame of the SAFECARE project, the IT threat detection system, which will be implemented, will be interconnected with the advanced file analysis system and the cyber threat monitoring system. Combining an IDS, a machine learning module and file analysis, all finely tuned, the IT threat detection system is meant to be unique and specific to SAFECARE. Requirements are met with the functionalities of the solution, however the innovative functionalities specified in this document need to be implemented since they do not exist yet.

The setup of the machine learning algorithm and the development of the machine learnings algorithms are the next steps of Task 5.1. It will be necessary to train the algorithms in a simulated environment. Thanks to the simulation platform, it will be possible to represent the information system of each end-user infrastructure. Attacks will be performed on the simulated information system to create gold datasets and to drive the algorithms with these datasets. Then the algorithms will be trained with real traffic data on each pilot site and finally the IT threat detection system will show its performance detecting the threats, when performing threat scenarios, during the demonstration phase.

References

SAFECARE internal references

[1] EUROPEAN COMMISSION Research Executive Agency. Grant Agreement number: 787002 — SAFECARE — H2020-CIP-2016-2017/CIP-2016-2017-2. 2019. Annex 1 (part A) — Innovation action.
[2] UG. *Initial requirements analysis — Deliverable 3.4*. V1.0, 28 February 2019.
[3] KUL. *Analysis of ethics, privacy, and confidentiality constraints — Deliverable 3.9*. V1, 28 February 2018.

Other references

[4] THE EUROPEAN PARLIAMENT AND THE COUNCIL OF THE EUROPEAN UNION. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), OJ L 119*, 4 May 2016.
[5] European Court of Human Rights — Council of Europe. *European Convention on Human Rights*, 4 November 1950.
[6] GDPR (4), article 5 (b)
[7] GDPR (4), article 5 (f)
[8] V. Jyothsna and V. V. Rama Prasad, "A Review of Anomaly based IntrusionDetection Systems," *International Journal of Computer Applications,* vol. Volume 28, no. 17, p. 0975 – 8887, 2011.
[9] P. M. Comar, . L. Liu, S. Saha, P.-N. Tan and A. Nucci, "Combining Supervised and Unsupervised Learning for Zero-Day Malware Detection," *IEEE INFOCOM,* 2013.
[10] S. Mukkamala, A. Sung and A. Abraham, "Cyber Security Challenges: Designing Efficient Intrusion Detection," pp. 125-161, 2005.
[11] A. Shiravi, H. Shiravi, M. Tavallaee and A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security,* 2012.
[12] C. I. f. Cybersecurity, "Intrusion Detection Evaluation Dataset (CICIDS2017)," 2018. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html.
[13] G. James, D. Witten, T. Hastie and R. Tibshirani, "An Introduction to Statistical Learning: With Applications in R," 2014.
[14] T. Hastie, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," *The Elements of Statistical Learning,* 2009.
[15] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001.
[16] Y. Freund and E. R. Schapire, "A Short Introduction to Boosting," *Journal of Japanese Society for Artificial Intelligence,* vol. 14, pp. 771-780, 1999.
[17] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen and W. Ma, "Lightgbm: A highly efficient gradient boosting," *Advances in Neural Information Processing Systems.,* pp. 3146-3154, 2017.
[18] S. M. e. S. L. Shah, "Fault detection and diagnosis in process data using one," *Journal of Process Control,* 2009.
[19] G. Giacinto, R. Perdisci, M. D. Rio and F. Roli, "Intrusion detection in computer networks by a modular ensemble of one-class classifiers," *Information Fusion,* 2008.
[20] M. Usama, J. Qadir, A. Raza, H. Arif, K.-}. A. Yau, Y. Elkhatib, A. Hussain and A. I. Al-Fuqaha, "Unsupervised Machine Learning for Networking: Techniques, Applications," *IEEE Acccess,* 2019.

[21] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research,* vol. 3, pp. 1157-1182, 2003.
[22] S. Gu, Y. Tan and X. He, "Recentness biased learning for time series forecasting," *Information Sciences,* vol. 237, pp. 29-38, 2013.