Ref. Ares(2021)1951157 - 18/03/2021

D4.3 PRELIMINARY VERSION OF CPSOS RUNTIME SECURITY MONITORING APPROACHES

Authors	Beatriz Gallego-Nicasio Crespo, …
Work Package	WP4 CPSoSaware System Layer Design and adaptation of dependable CP(H)SoS

Abstract

This document is a report that describes the demonstrator of the preliminary Security Runtime Monitoring and Management of CPSoSAware. The SRMM is a subsystem of the CPSoSAware architecture, in charge of monitoring the CPSs and the infrastructure that enables their inter-communication and control, with the objective of detecting and warning about any suspicious and malicious activity that may threaten the overall security properties of the system.

Funded by the Horizon 2020 Framework Programme
of the European Union

Deliverable Information

Work Package	WP4 CPSoSaware System Layer Design and adaptation of dependable CP(H)SoS
Task	T4.3 [M6-M28] CPSoSAware Security Runtime monitoring and Management (SRMM) Design and Development.
Deliverable title	D4.3 Preliminary Version of CPSoS Runtime Security Monitoring Approaches
Dissemination Level	PU
Status	D: Draft
Version Number	0.6
Due date	28/02/2021

Project Information

Project start and duration	01/01/2020 – 31/12/2022, 24 months
Project Coordinator	Industrial Systems Institute, ATHENA Research and Innovation Center 26504, Rio-Patras, Greece
Partners	1. ATHINA-EREVNITIKO KENTRO KAINOTOMIAS STIS TECHNOLOGIES TIS PLIROFORIAS, TON EPIKOINONION KAI TIS GNOSIS (ISI) the Coordinator 2. FUNDACIO PRIVADA I2CAT, INTERNET I INNOVACIO DIGITAL A CATALUNYA (I2CAT), 3. IBM ISRAEL - SCIENCE AND TECHNOLOGY LTD (IBM ISRAEL 4. ATOS SPAIN SA (ATOS), 5. PANASONIC AUTOMOTIVE SYSTEMS EUROPE GMBH (PASEU) 6. EIGHT BELLS LTD (8BELLS) 7. UNIVERSITA DELLA SVIZZERA ITALIANA (USI), 8. TAMPEREEN KORKEAKOULUSAATIO SR (TAU) 9. UNIVERSITY OF PELOPONNESE (UoP) 10. CATALINK LIMITED (CATALINK) 11. ROBOTEC.AI SPOLKA Z OGRANICZONA ODPOWIEDZIALNOSCIA (RTC) 12. CENTRO RICERCHE FIAT SCPA (CRF) 13. PANEPISTIMIO PATRON (UPAT)
Website	www.cpsosaware.eu

Control Sheet

VERSION	DATE	SUMMARY OF CHANGES	AUTHOR
0.1	17/12/2020	Initial Draft circulated to the Consortium	Beatriz Gallego-Nicasio Crespo (ATOS)
0.2	19/02/2021	Second Draft (incomplete) shared with the Consortium	Beatriz Gallego-Nicasio Crespo (ATOS)
0.3	26/02/2021	Third Draft (incomplete) shared with the Consortium	Beatriz Gallego-Nicasio Crespo (ATOS)
0.4	09/03/2021	Final complete draft, ready for internal review	Beatriz Gallego-Nicasio Crespo (ATOS)
0.5	15/03/2021	Final version, addressing comments from internal review	Beatriz Gallego-Nicasio Crespo (ATOS)
0.6	17/03/2021	Final version for submission	Beatriz Gallego-Nicasio Crespo (ATOS)

	NAME
Prepared by	ATOS
Reviewed by	IBM, I2CAT
Authorised by	

DATE	RECIPIENT
17/03/2021	Project Consortium
30/06/2020	European Commission

Table of contents

List of tables

List of figures

Executive summary

This document describes the demonstrator of the preliminary Security Runtime Monitoring and Management (SRMM) of CPSoSAware. The SRMM is a subsystem of the CPSoSAware architecture, in charge of monitoring the CPSs and the infrastructure that enables their inter-communication and control, with the objective of detecting and warning about any suspicious and malicious activity that may threaten the overall security properties of the system.

The SRMM monitors the CPSoS landscape at two levels: at each individual CPS level and at higher system level (involving all the communications between CPSs and between the CPS and the cloud). In order to do this, firstly we have been identified the assets that compose this complex environment, including CPS sensors, intra-communication devices, control systems, inter-communication elements and the virtual infrastructure that enables the correct, efficient and safe functioning of a CPSoS realm. Secondly, the threats/attacks landscape has been analysed, to identify the security runtime and detection capabilities that are required for the SRMM. Two levels of security analysis were performed: at the security sensor level and at the Security Information and Event Management (SIEM) level. The first one performs a thorough and very specific analysis of the data processed, exchanged and maintained in each monitored asset, as well as of the processes and resources managed by them. This permits identifying anomalous behaviours and suspicious activity happening at the monitored asset, and generates the corresponding security events. At SIEM level, these events are correlated with additional context information to detect potential security incidents and warn of the risk and impact associated to them.

Considering these capabilities of the SRMM, the requirements of the CPSoS ecosystem and the CPSoSAware general architecture described in D1.3[11], a first version of the SRMM architecture is proposed. This architecture foresees a two-layered hierarchical SRMM: one layer devoted to the security monitoring of the CPSs and reporting to a high-level layer that monitors and manages the security of the CPSoS from a global perspective. The SRMM architecture defines interfaces to communicate with other components of the CPSoSAware architecture, namely the Task 3.5 CPS sensors/agents, the Task 2.2 and Task 4.2 System Inter-communication layer components and Task 2.1 Data Collection (CSAIE) component.

The implementation of the SRMM architecture relies to a great extent in the XL-SIEM technology of Atos. The XL-SIEM is a SIEM solution with added high-performance correlation engine to deal with large volumes of security information. It provides scalability and distribution in security events processing through a cluster of nodes, and capacity to raise security alerts from a business perspective based on events collected from different data sources at different layers. This SIEM technology is complemented with the security runtime monitoring capabilities of the security sensors and agents deployed at the different layers of the CPSoS infrastructure.

An example demonstration scenario is included in this document, to illustrate the deployment and use of the preliminary version of the SRMM demonstrator to monitor and detect an attempt to perform a Distributed Denial of Service attack in a simplified connected autonomous vehicle context.

1 Introduction

This document describes a demonstrator of the preliminary version of the Security Runtime Monitoring and Management (SRMM) of CPSoSAware. The document describes the functionalities, the architecture and the first version of the technology that implements the SRMM, which is based on the XL-SIEM technology of Atos.

1.1 Document structure

This document is structured into five sections:

- **Section 1** corresponds to this introduction.
- **Section 2** presents the security runtime monitoring and threat detection functionality in the context of CPSoSAware.
- **Section 3** is a description of the first version of the SRMM architecture.
- **Section 4** describes the SRMM demonstrator – preliminary version, including a description of the XL-SIEM technology and a demonstration scenario to illustrate its use in the context of the Autonomous vehicle use case scenario.
- **Section 5** concludes the document and outlines the next steps.

2 Security Runtime Monitoring and Threat Detection

This section describes from a general perspective, the scope of the Security Runtime Monitoring and threat detection functionality in the CPSoS context.

2.1 Threats and attacks landscape

Deliverable D1.1[1] reviewed, in section 7.2, the threat models, taxonomies and attack classifications proposed by relevant projects and initiatives of relevance for the different CPSoSAware architectural domains: system, communication and device. The following table summarizes the work presented in D1.1 section 7.2.

Domain	Related Asset / Component	Threats / Attacks
System		Physical attack (deliberate/ intentional) - Fraud Physical attack (deliberate/ intentional) - Sabotage Physical attack (deliberate/ intentional) - Vandalism Physical attack (deliberate/ intentional) - Theft (devices, storage media and documents) Physical attack (deliberate/ intentional) - Information leakage/sharing Physical attack (deliberate/ intentional) - Unauthorized physical access / Unauthorised entry to premises Physical attack (deliberate/ intentional) - Coercion, extortion or corruption Physical attack (deliberate/ intentional) - Damage from the warfare Physical attack (deliberate/ intentional) - Terrorists attack Unintentional damage / loss of information or IT assets - Information leakage/sharing due to human error Unintentional damage / loss of information or IT assets - Erroneous use or administration of devices and systems Unintentional damage / loss of information or IT assets - Using information from an unreliable source

		Unintentional damage / loss of information or IT assets - Unintentional change of data in an information system Unintentional damage / loss of information or IT assets - Inadequate design and planning or improperly adaptation Unintentional damage / loss of information or IT assets - Damage caused by a third party Unintentional damage / loss of information or IT assets - Damages resulting from penetration testing Unintentional damage / loss of information or IT assets - Loss of information in the cloud Unintentional damage / loss of information or IT assets - Loss of (integrity of) sensitive information Unintentional damage / loss of information or IT assets - Loss of devices, storage media and documents Unintentional damage / loss of information or IT assets - Destruction of records Disaster (natural, environmental) - Disaster (natural earthquakes, floods, landslides, tsunamis, heavy rains, heavy snowfalls, heavy winds) Disaster (natural, environmental) - Fire Disaster (natural, environmental) - Pollution, dust, corrosion Disaster (natural, environmental) - Thunder stroke Disaster (natural, environmental) - Water Disaster (natural, environmental) - Explosion Disaster (natural, environmental) - Dangerous radiation leak Disaster (natural, environmental) - Unfavourable climatic conditions Disaster (natural, environmental) - Major events in the environment Disaster (natural, environmental) - Threats from space / Electromagnetic storm Disaster (natural, environmental) - Wildlife

		Failures/ Malfunction - Failure of devices or systems Failures/ Malfunction - Failure or disruption of communication links (communication networks) Failures/ Malfunction - Failure or disruption of main supply Failures/ Malfunction - Failure or disruption of service providers (supply chain) Failures/ Malfunction - Malfunction of equipment (devices or systems) Outages - Loss of resources Outages - Absence of personnel Outages - Strike Outages - Loss of support services Outages - Internet outage Outages - Network outage Eavesdropping/ Interception/ Hijacking - War driving Eavesdropping/ Interception/ Hijacking - Intercepting compromising emissions Eavesdropping/ Interception/ Hijacking - Interception of information Eavesdropping/ Interception/ Hijacking - Interfering radiation Eavesdropping/ Interception/ Hijacking - Replay of messages Eavesdropping/ Interception/ Hijacking - Network Reconnaissance, Network traffic manipulation and Information gathering Eavesdropping/ Interception/ Hijacking - Man in the middle/ Session hijacking Nefarious Activity/ Abuse - Identity theft (Identity Fraud/ Account) Nefarious Activity/ Abuse - Receive of unsolicited E-mail

		Nefarious Activity/ Abuse - Denial of service Nefarious Activity/ Abuse - Malicious code/ software/ activity Nefarious Activity/ Abuse - Manipulation of information Nefarious Activity/ Abuse - Misuse of audit tools Nefarious Activity/ Abuse - Misuse of information/ information systems (including mobile apps) Nefarious Activity/ Abuse - Unauthorized activities Nefarious Activity/ Abuse - Unauthorized installation of software Nefarious Activity/ Abuse - Compromising confidential information (data breaches) Nefarious Activity/ Abuse - Hoax Nefarious Activity/ Abuse - Remote activity (execution) Nefarious Activity/ Abuse - Targeted attacks (APTs etc.) Nefarious Activity/ Abuse - Failed of business process Nefarious Activity/ Abuse - Brute force Nefarious Activity/ Abuse - Abuse of authorizations Legal - Violation of laws or regulations / Breach of legislation Legal - Failure to meet contractual requirements Legal - Unauthorized use of IPR protected resources Legal - Abuse of personal data Legal - Judiciary decisions/court orders
Communication	SDN networks	Spoofing attacks Main in the middle attacks Tampering Repudiation

		Information disclosure Denial of Service – Flooding and Saturating attacks
	Wireless Sensor Networks (WSN)	Passive attack – Data Interception – Traffic Analysis Passive attack – Data Interception – Sniffing Passive attack – Data Interception – Key logger Active attack - Packets crafting – Replay attack Active attack - Packets crafting – Masquerading Active attack - Packets crafting – 0-day Active attack - Packets alteration – Main-in-the-middle (MiM) Active attack – Service compromising – DoS Active attack – Service compromising – DdoS Active attack – Service compromising – SQL Injection
	V2X Communication	DDoS attacks, doxing, website defacements Information theft, virtual sabotage, website parodies Whistleblowing Gathering information about network (reconnaissance) Man in the middle (MiM) Session hijacking Repudiation of actions Use crimeware, phishing, and spear-phishing Trojan Smash-and-grab, social engineering, business email compromise (BEC) scams, botnets, password attacks, malware, ransomware Interception of information

		Replay of messages Account hijacking Network reconnaissance Data exfiltration or privilege misuse Spear-phishing password attacks, social engineering, direct compromise, data exfiltration, remote access trojans, and destructive malware. Interfering radiation Cyber reconnaissance of critical infrastructure Defacements and claimed leaks Worm Spoofing
CPS	Device perception layer	Sensor Alteration, Data theft Sensitive information leakage Denial of Service Physical Attack on a Device
	Device Application layer	Malformed Firmware/Hardware Integrity Attacks Against Machine Learning Logging Mechanism alteration Application software functionality change

ENISA recently published a report that analyses the top cyber threats for the period January 2019-April 2020 [2] and describes several trends. According to this report, the top 15 cyberthreats in the period reviewed are: 1 - Malware, 2 - Web-based Attacks, 3 – Phishing, 4 - Web application attacks, 5 – Spam, 6 - Denial of service, 7 - Identity theft, 8 - Data breaches, 9 - Insider threat, 10 – Botnets, 11 - Physical manipulation, damage, theft and loss, 12 - Information leakage, 13 – Ransomware, 14 – Cyberespionage and 15 – Cryptojacking. More specifically, the Sectorial / Thematic Threat Analysis published by ENISA[3] analyses trends in reported incidents by sector, including relevant ones such as Information and Communication or Manufacturing. In both cases, the most popular attacks are related to these threats: Web application attacks, Insider threat (unintentional abuse/error) and Malware. The report also analyses trends in threats affecting technologies of special relevance for the CPSoSAware context such as Next generation of mobile communications or 5G, Internet-of-things (IoT) or Smart Cars. For each of these technologies, the report lists relevant threats grouped by related asset or component group.

2.2 Run-time Security Monitoring in the CPSoS context

Deliverable D1.1 introduced the concept of run-time security monitoring and identified the main components that should be part of it, namely: Intrusion Detection Systems (IDS), malware detectors and anomaly detectors, all of them connected and acting as source of security information for a Security Information and Event Management (SIEM) system. These elements should be tailored to the specificities of the technologies that compose a CPSoS system, but at the end of the day, two main functions should be provided:

- Collection of data (security-related information) from heterogeneous data sources deployed and monitoring each domain and layer of the CPSoS
- Analysis of the information collected from a security perspective, in order to detect anomalies, vulnerabilities, or security incidents.

The CPSoSAware ecosystem model presented in D1.1 section 7.2 identified three major domains, namely System domain, Communication domain and CPS device domain; which should be adequately monitored with specific components named security monitoring agents and sensors. These sensors and agents inspect configurations, data and behaviour of the elements of each domain to identify changes, detect unusual or suspicious behaviour and, in some more advanced cases, analyse these findings to report security events. Security events are correlated and analysed with the support of SIEMs, to detect threats and complex attack scenarios that involve different steps or actions taken by attackers at different levels in the system, in order to accomplish their malicious objective. When a series of security events collected from the monitored infrastructure matches a threat or attack pattern, SIEMs generate security alarms that inform the security operators of a potential security incident happening. Security alarms are evaluated by security incident handling teams to decide whether or not start an investigation, trigger automatic remediation or mitigation processes and also, are used to compute metrics that provide a view of the security situation of a system.

Security agents and sensors monitor the physical, virtual and software elements that integrate a CPSoS system. As already explained, these components report information, in the form of security events, to the SIEM system that processes these events, correlating and performing a security analysis that may result in security alarms. But security alarms can also be fed into a SIEM system, as any other security event, for cross-correlation of complex multi-level security run-time monitoring. In this way, a SIEM can be considered another type of security sensor that reports to a higher-level SIEM. This approach can be selected when we want to have specialized SIEMs monitoring a very specific sub-system and there are resource constraints that prevent from deploying a complete SIEM in the sub-system, as it may be the case of CPS systems.

Depending on the type of threats and attacks monitored and on the technological characteristics of the monitored infrastructure domain, in CPSoSAware we consider the following security runtime monitoring capabilities:

- CPS-level security monitoring: consists of observing the activity of the system within each individual CPS to collect security-relevant events, correlate them and generate information that serves to understand the security situation of the CPS at any point in time, and take local actions if needed.

In this context, we can distinguish the following security monitoring capabilities, each one focused on a CPS architectural layer:

  - *Monitoring of the physical/device layer*: with a focus on the status and behaviour of the sensing and actuation devices of the CPS, e.g., GPS, Lidar, etc.
  - *Monitoring of the applications*: this capability focuses on monitoring security aspects related to the configuration and activity of the software services, applications and business processes running in the CPS.
  - *Monitoring of the data*: the objective in this case is to monitor security properties of the data stored, processed, and transmitted within the boundaries of the CPS.
  - *Monitoring of the intra-communication*: this capability monitors the communication interfaces of the CPS to detect anomalies in the intra-communication behaviour.

- System-level security monitoring: consist of observing the system as a whole, collecting security-related information from all the elements that compose the Cyber Physical System of Systems and correlating them at the same level. Thus, each individual CPS that belongs to the CPSoS is considered as another element of the system that is subject to fail or be attacked and because of that, security information (i.e., security alarms) is collected from CPSs and processed at system-level. By doing this, it is possible analysing the security situation of the individual CPSs from a global perspective and detect more complex security anomalies. Besides this, the system-level security monitoring has the following capabilities:
  - *Monitoring of the virtual/physical layer:* this capability focuses on monitoring security aspects related to the virtual and physical infrastructure that supports the CPSoS.
  - *Monitoring of the application layer*: in this case the focus is on monitoring security aspects of the Cloud applications and services that permit operating, controlling and orchestrating the CPSoS. Similarly to the CPS-level monitoring, configuration files and activity logs are the main asset to be monitored in this case.
  - *Monitoring of the data layer:* refers to monitor security properties of the data stored and processed by the Cloud applications and services used to operate, orchestrate and control the CPSs, including the data exchanged between the controller and the CPSs and the data exchanged between CPSs.
  - *Monitoring of the inter-communication layer:* this capability focuses on monitoring security aspects related to the communications between the system and the edge nodes or CPSs.

2.3 Security analysis for detection of anomalies and threats

As already introduced in the previous section, the security analysis for detection of threats and attacks can occur not only at the system level, i.e., performed by SIEMs, but also at the level of sensors and agents. This is because some sensors and agents are not just a probe that collects and reports raw data but have advanced capabilities for processing and analysing security-related information and produce security events. In the following, we discuss the security analysis capabilities at two levels: at the sensor level and at the SIEM level.

2.3.1 Security analysis performed at sensor level

Security monitoring sensors are pieces of software that observe a category of asset of the infrastructure, such as data or network communications, in order to collect specific information and possibly searching for certain pattern match. The information collected is processed and the result is generated as an output that can be logged into a file or displayed in an output interface. In CPSoSAware, security monitoring sensors have capabilities to process the information collected about the observed asset and perform a security analysis that generates, as a result, security events. Task 3.5 “Security Sensors and Agents” is devoted to the research and development of this topic, but here are listed a preliminary list of security sensor categories used in CPSoSAware:

Table 1 CPSoSAware Security Monitoring Sensors

Security Sensor	Capabilities/Description	Monitored Layer / Asset Category	Security Events generated
Suricata (https://suricata-ids.org/)	Network Intrusion Detection System (NIDS) engine Network Intrusion Prevention System (NIPS) engine Network Security Monitoring (NSM) engine Offline analysis of PCAP files Traffic recording using pcap logger Unix socket mode for automated PCAP file processing Advanced integration with Linux Netfilter firewalling	Communication, both at System and CPS level. Support for packet decoding of: IPv4, IPv6, TCP, UDP, SCTP, ICMPv4, ICMPv6, GRE Ethernet, PPP, PPPoE, Raw, SLL, VLAN, QINQ, MPLS, ERSPAN, VXLAN, Geneve App layer decoding of: HTTP, HTTP/2, SSL, TLS, SMB, DCERPC, SMTP, FTP, SSH, DNS, Modbus, ENIP/CIP, DNP3, NFS, NTP, DHCP, TFTP, KRB5, IKEv2, SIP, SNMP, RDP, RFB, MQTT	Thousands of different security events grouped into categories and sub-categories, e.g. Suspicious – Network activity, Exploit – SQL Injection, Suspicious Scada Activity, Malware – Trojan, Recon - Scanner, etc.
OSSEC (https://ossec.net)	Host-based Intrusion Detection System (HIDS) Features: Log analysis, file integrity monitoring, Windows registry monitoring, centralized policy enforcement, rootkit detection, real-time alerting and active	Applications and Data, both at System and CPS level. Monitor Integrity of Files and Logs from systems, devices and applications	Hundreds of different security events grouped into categories and subcategories, e.g. Authentication, System Information,

	response. It runs on most operating systems, including Linux, OpenBSD, FreeBSD, MacOS, Solaris and Windows.		Inventory change, etc.
Kismet (www.kismetwireless.net)	Kismet is a wireless network and device detector, sniffer, wardriving tool, and WIDS (wireless intrusion detection) framework.	Wi-Fi interfaces, Bluetooth interfaces, some SDR (software defined radio) hardware like the RTLSDR, and other specialized capture hardware.	Various events such as: Possible ap spoofing channel change Suspicious traffic Suspicious client Flood detected
CPS Hardware Security Token	The sensor monitors the data integrity of exchanged messages between CPS The sensor identifies possible spoofing of the car GPS sensor. The accurate position of the GPS is calculating by fusing other modalities on the CPS Design space exploration of different integrity check techniques Computation monitors to detect anomalous processing activity (such as CPU load….)	Data at CPS level. TCP, UDP, HTTP/HTTPs CAN bus, deployed firmware GPS sensor data Computation	Integrity failure Authentication Failure GPS Spoofing GPS unavailability Excessive resource usage

CPS Communication Security Integrity	The sensor monitors if the communication interface identified that the communicating interface attempting to connect doesn't have the correct credentials (BLE, WiFi, ZigBee applicable)	Intra-CPS communication. BLE: Pairing credentials, WiFi: SSID/Password/MAC address, ZigBee: MAC address	WIFI authentication failure BLE authentication failure ZigBee authentication failure
CPS Communication Health status	Monitors if the communication interface identifies communicating link failure	Intra-CPS communication. In communication scenarios that responses are expected (bidirectional), they are not received after a specific time delay. In communication scenarios that responses are not expected (uni-directional), ping-like services can be deployed to monitor the link status	WIFI link failure BLE link failure ZigBee link failure
CPSoS authentication	Securely identify each individual CPS/CPHS system (e.g., ADAS Car, AGV, etc.)	Device/Edge at System-level. Identification/authentication token	Edge server unavailability Authentication Failure with Edge Server

2.3.2 Security analysis performed at SIEM level

SIEM systems collect information about a monitored complex infrastructure through the use of software agents, which are usually deployed at specific elements of the infrastructure that have access to the sources of information that are being monitored: network traffic, application logs, databases, etc. The information collected by agents is parsed, normalized and encapsulated in the form of events that follow a specific data format. Events are sent to the SIEM server for correlation, using predefined security directives or rules, in order to identify anomalous behaviours, discover possible threats and detect security incidents. When a specific set of events received matches a directive, a security alarm is raised and this, in turn may trigger actions according to predefined policies. Security alarms usually contain information about the threat or security anomaly detected, the affected infrastructure asset and the source of the security event (e.g. the source IP of an attack performed from an external actor). Besides that, security alarms may also contain information to determine the severity of the incident, such as reliability of the information collected, or risk associated to the asset affected by the incident. This type of information can be used to take adequate and proportionate actions to address or mitigate the incident. Some examples of these actions are notifying the security administrator (through email, dashboard, etc.) or the automatic or semi-automatic execution of certain reactions to reconfigure the system or implement more specific countermeasures.

SIEMs can be classified according to their features: data sources supported, data storage capabilities, processing capabilities, flexibility of the security directives, support for behavioural analysis, support for risk analysis, extensibility and interoperability through available APIs, resilience, visualization capabilities, reaction capabilities, deployment model, scalability or licensing, among others. Other advanced capabilities of SIEMs are support for forensics and threat hunting, cloud readiness or support for advanced threat detection and response. Research and advisory IT organisations, such as Gartner[4], Forrester or [5]TechTarget[6], compare, classify, and evaluate SIEMs considering other business and market-related aspects too. But overall, SIEMs implement a general concept, which is depicted in Figure 1. In this figure, the different monitored infrastructure realms are depicted as sources of data of different nature at the bottom, communicating with the SIEM server through SIEM agents. At the top of the figure, the SIEM server stores the events collected from the monitored infrastructure and alarms triggered in a database and hosts the correlation engine and security intelligence processes that permit security administrators have an overview of the security situation of the monitored system at any point in time.

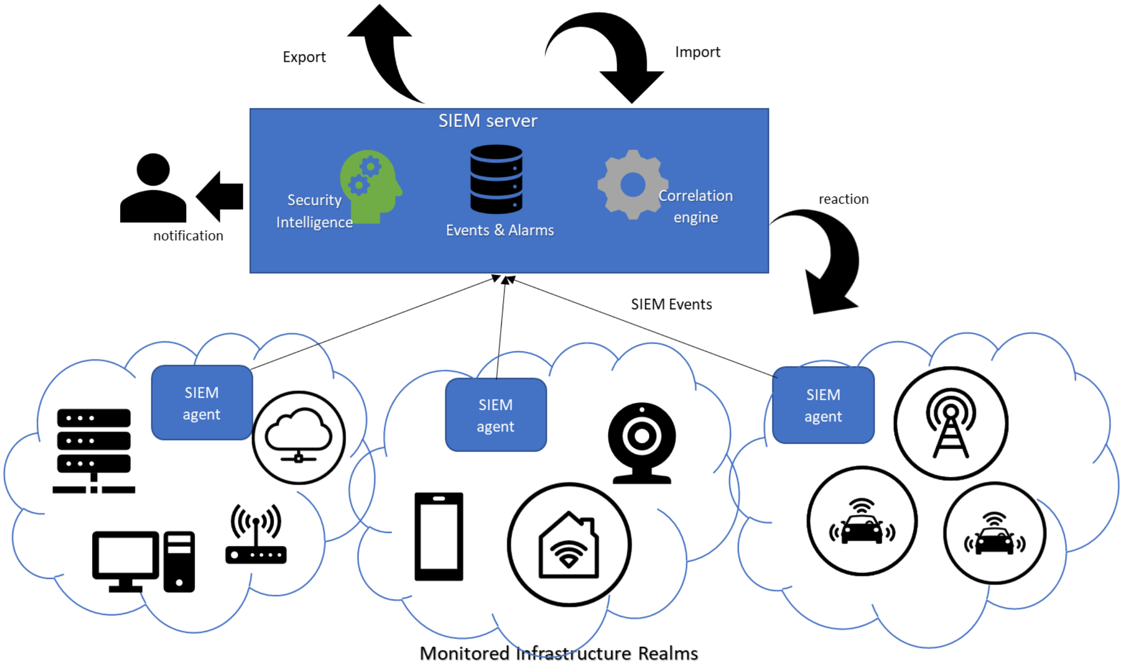


Figure 1 SIEM concept

SIEMs support the security analysis through different tasks. At design-time, prior to the deployment of the security runtime monitoring infrastructure, the following activities should be done to adapt the SIEM security analysis to the specifics of the monitored CPSoS context:

- **Identification and characterisation of the infrastructure assets**: considering the complexity of a CPSoS and the different nature of the elements that compose it, the first step is the identification of the assets that should be monitored and their characterisation, according to technical, business and security-related criteria. This activity permits establishing what are the critical assets in the system, with a higher business or security value, and design appropriate security directives and policies to protect them.
- **Identification and characterisation of data sources:** once the elements of the infrastructure that should be monitored have been identified, it is necessary to analyse the information that can be collected from them and select what is relevant from a security perspective. The result of this analysis is a list of data sources and event types, associated to these data sources, that will constitute the input for the correlation processes performed at the SIEM server.
- **Design and implementation of security directives:** this activity consist of analysing the characteristics of the CPSoS, the security requirements and the threats that may affect the system, in order to define possible attack/threat scenarios, suspicious or anomalous situations that should be monitored. These scenarios are translated into event patterns, which capture relationships of different type (temporal, causal) between events. Event patterns are codified as security rules or directives and are used by the correlation engine of the SIEM server to detect occurrence of such patterns in the events that are being collected at run-time from the monitored environment.

At run-time, once the sensors, agents and SIEM server are deployed and running, the real-time analytics are done at two levels:

- **Real-time processing of security events**: consist of performing computing operations on the events received from the environment. Events can be received as streams or continuous flows and that is referred in literature as event stream processing. There are different platforms in the market that perform Data/Event Stream Processing, such as Hadoop, Spark, Storm, Kafka, Flume or Amazon Kinesis. As it is described in section 4.1.3, the technology that implements the runtime security analysis of the CPSoSAware SRMM is based on Apache Storm.
  The operations performed on events consist in filtering, aggregating, and correlating multiple events coming from the same or different sources, resulting in complex event computations, often referred as Complex Event Processing (CEP). These operations are executed in the SIEM server by the correlation engine, in accordance with the correlation rules contained in the security directives defined at design-time. Event processing systems can be classified into those that follow a query-based approach, a rule-oriented approach or a programmatic approach. As it is described in section 4.1.3, in CPSoSAware it is used a correlation engine technology that follows a query-based approach.
- **Analysis of security alarms**: alarms contain multiple useful information about the security anomaly, incident or threat detected, about the affected asset, but also contextual information of the environment around the detection that can be used for statistical analysis to identify trends and implement predictive algorithms. Moreover, the analysis of security alarms combined with forensic techniques can trace back until the root cause of the incident. Alarms usually contain reliability and risk values, which combined with the criticality of the affected asset can be used to perform an assessment of the severity of the incident and evaluate the impact in the overall security posture of the system. Last but not least, alarms can be exported into standard formats such as MISP or STIX, and feed third-party Threat Intelligence Analysis platforms or SOCs for further analysis and this way, contribute to the cybersecurity community.

2.4 Reporting

Security runtime monitoring sensors, in combination with SIEMs, collect and produce security-related information from a target monitored system and this information can be used for different purposes. As it is explained in section 2.3, security events and alarms help security administrators to be aware of security anomalies or incidents that may be happening in the infrastructure and respond to them promptly and adequately. But this information can also be used to compute security metrics that CISOs can use to assess the security posture of a complex system and take adequate corrective actions.

In a CPSoS, the assessment of the security of the system can be done both at the individual CPS level and at the general system level. Reporting security information at CPS level permit evaluating whether the specific requirements of the CPS are met or not and thus, take adequate corrective actions and reconfigurations locally, in a fast and suitable manner. On the other hand, evaluating the overall situation of the system, considering each CPS from a global perspective, provide a more comprehensive understanding of all the factors that may influence the achievement of the system security goals, and apply general corrective actions that fit all the possible situations.

Task 2.5 defines security metrics that will be computed on the information contained in security alarms and events produced by the Security Runtime Monitoring component of CPSoSAware. This is a preliminary set of metrics defined, which will be further developed in the corresponding Task 2.5 deliverable:

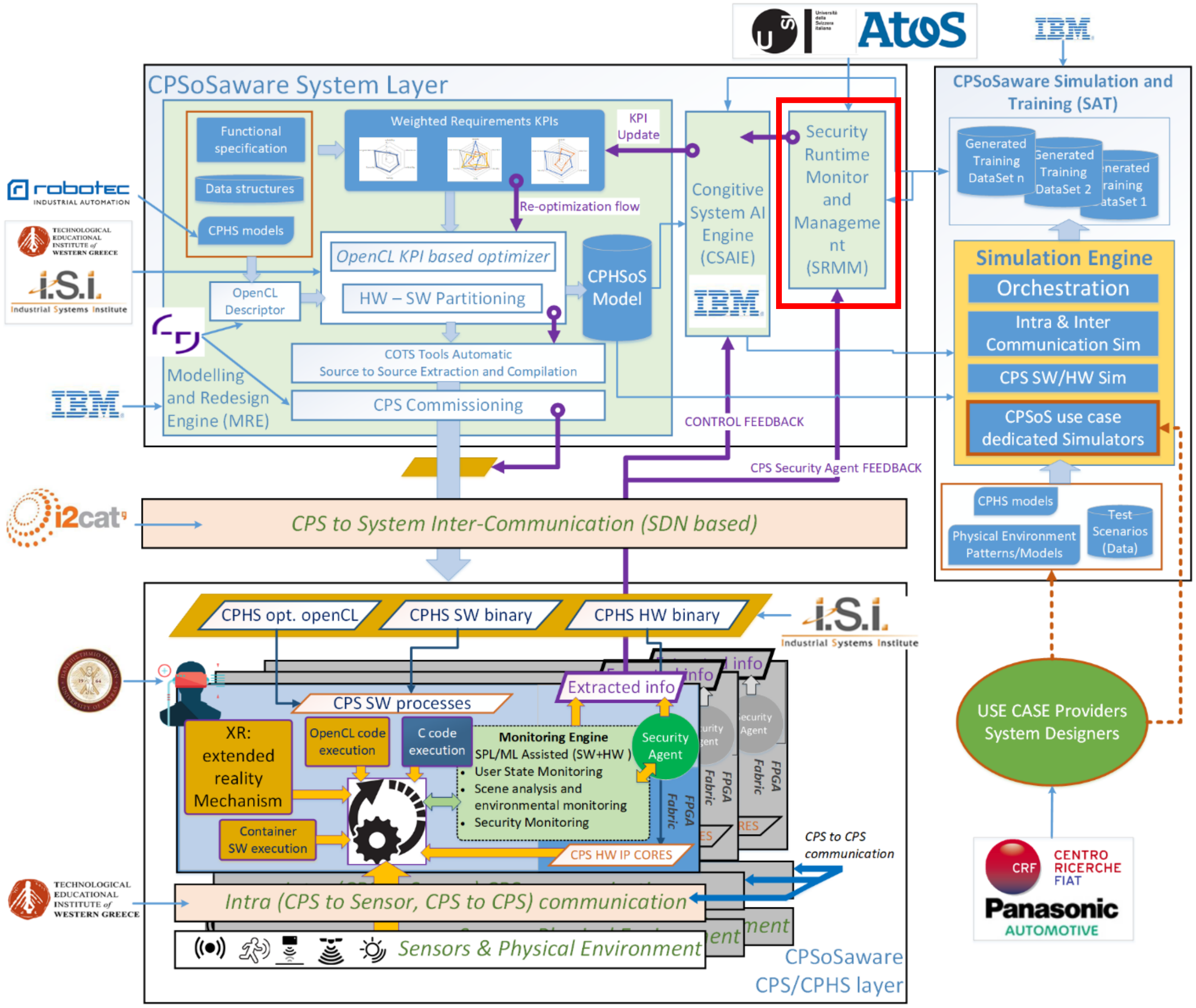
- *Assessment of the severity of a detected security issue in the system*, based on the DREAD methodology[7] to rate, compare and prioritize the severity of the risk presented by a security issue detected in the system.
- *Assessment of the impact that a detected security issue has in the system*, based on the STRIDE methodology[8] to assess the impact that a detected security issue has on the security properties of each asset of the system.
- *Criticality of a detected security issue*, based on the methodology defined in H2020 project ANASTACIA, measures the criticality of a detected security issue as a combination of the severity of the issue and the impact it has in the requirements of the system[9].
- *Risk associated to a detected security issue*, based on the OWASP Risk Rating Methodology[10]: Risk=Likelihood * Impact

3 Security Runtime Monitoring and Management (SRMM): preliminary architecture design

This chapter describes the preliminary version of the architecture of the SRMM of CPSoSAware.

3.1 Positioning of the SRMM in the CPSoSAware architecture and requirements

The Security Runtime Monitoring and Management is a sub-system of the system layer of the CPSoSAware architecture. Figure 2 highlights with a red rectangle the SRMM in the general view of the CPSoSAware architecture, as a module that receives feedback from CPS-level security agents (purple arrow labelled as “CPS Security Agent FEEDBACK”) and produces output for the Cognitive System AI Engine (CSAIE).



CPSoSaware System Layer
Functional specification
Data structures
CPHS models
Weighted Requirements KPIs
KPI Update
Re-optimization flow
OpenCL KPI based optimizer
HW – SW Partitioning
OpenCL Descriptor
CPHSoS Model
COTS Tools Automatic Source to Source Extraction and Compilation
CPS Commissioning
Modelling and Redesign Engine (MRE)
Congitive System AI Engine (CSAIE)
Security Runtime Monitor and Manageme nt (SRMM)
CONTROL FEEDBACK
CPS Security Agent FEEDBACK
CPSoSaware Simulation and Training (SAT)
Generated Training DataSet n
Generated Training DataSet 2
Generated Training DataSet 1
Simulation Engine
Orchestration
Intra & Inter Communication Sim
CPS SW/HW Sim
CPSoS use case dedicated Simulators
CPHS models
Physical Environment Patterns/Models
Test Scenarios (Data)
CPS to System Inter-Communication (SDN based)
CPHS opt. openCL
CPHS SW binary
CPHS HW binary
Extracted info
CPS SW processes
XR: extended reality Mechanism
OpenCL code execution
C code execution
Container SW execution
Monitoring Engine
SPL/ML Assisted (SW+HW)
User State Monitoring
Scene analysis and environmental monitoring
Security Monitoring
Security Agent
FPGA Fabric
CPS HW IP CORES
CPS to CPS communication
Intra (CPS to Sensor, CPS to CPS) communication
Sensors & Physical Environment
CPSoSaware CPS/CPHS layer
USE CASE Providers System Designers

Figure 2 General view of the CPSoSAware architecture - the SRMM is highlighted with a red rectangle

Deliverable D1.3 [11] includes a technical component specification of the SRMM identifying the following functional and non-functional requirements:

- **TC4.3.1.R1 Input**: The component must receive normalized security events through TCP/41000 from agents/sensors deployed remotely, in the infrastructure that is under surveillance. Events comply with a predefined JSON format.
- **TC4.3.1.R2 Configuration**: The component should be configured using the component's graphical dashboard, to define the security monitoring infrastructure in use (topology of sensors/agents deployed and active), the security detection rules and the correlation directives.
- **TC4.3.1.R3 Events Processing**: The component must process security events received as input, correlate them using the security detection rules configured, and generate security alarms as ouput, as defined in the correlation directives configured.
- **TC4.3.1.R4 Output**: The component should produce as output security alarms. Alarms comply with a predefined JSON format. Alarms can be configured to be persisted in a DB, logged into a file, transmitted to a third-party component (using a middleware such as Message Queue/Broker) and displayed in the SRMM graphical dashboard.
- **TC4.3.1.R5 Cross-correlation**: Security alarms produced as output by the SRMM can be configured to be input into the SRMM correlation engine, for cross-correlation processes.
- **TC4.3.1.NFR1 Scalability**: of the SRMM correlation engine and data collection module
- **TC4.3.1.NFR2 High-performance**: of the SRMM correlation engine and the data persistence layer
- **TC4.3.1.NFR3 Integrity**: of the security events transmitted from sensors/agents to the SRMM component, and of the security alarms generated as output by the SRMM
- **TC4.3.1.NFR4 Confidentiality**: of the security events transmitted from sensors/agents to the SRMM component, and of the security alarms generated as output by the SRMM
- **TC4.3.1.NFR5 Accountability**: of the security events transmitted from sensors/agents to the SRMM component, of the correlation process and of the security alarms generated as output by the SRMM

The architecture of the SRMM, described in the next section, addresses all the above-listed requirements.

3.2 SRMM internal architecture description

The overall approach of Security Runtime Monitoring in CPSoSAware is depicted in Figure 3. The picture shows the interaction between Task 4.3 and other related tasks in the project, namely Task 3.5, Task 2.2 and Task 4.2, which provide input data to the SRMM component; and with Task 2.1, which will collect data reported by the SRMM for statistical analysis and metrics computation. In the figure, in the left hand-side, the configuration of the SRMM defines data sources, the topology of the infrastructure to be monitored and the security directives and policies to apply for the detection of anomalies and threats. Bottom up, the Events Collection is in charge of receiving input from CPS Sensors/Agents and the Inter-communication layer. Next, Events Processing applies filtering and aggregation rules before correlating the events according to the predefined security directives. Alarms generated by the correlation process are analysed and cross-correlated from a CPSoS perspective. The results of the analysis, as well as the security events and alarms are reported to the external CPSoSAware Data Collection module for further analysis and metrics computation. On the right hand-side of the figure is depicted the storage of security events, alarms and logs for accountability and support forensic analysis.

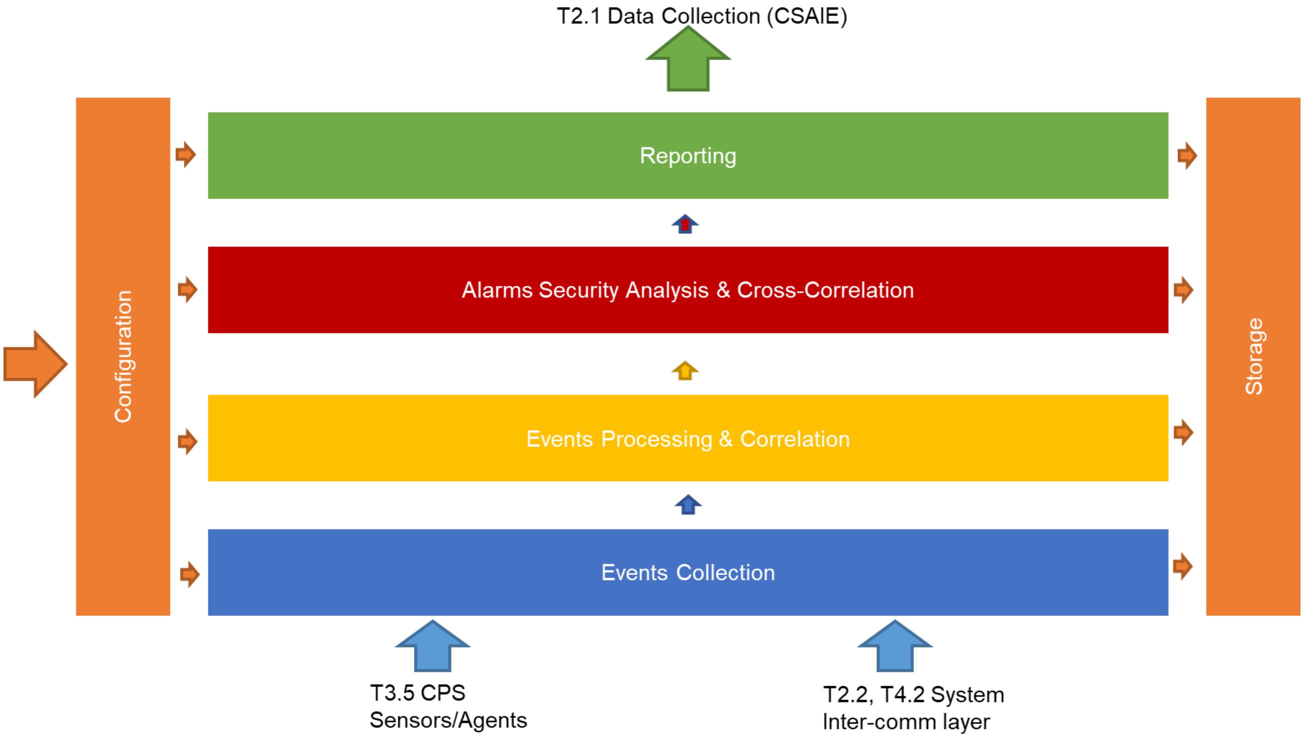

T2.1 Data Collection (CSAIE)
Configuration
Reporting
Alarms Security Analysis & Cross-Correlation
Events Processing & Correlation
Events Collection
Storage
T3.5 CPS Sensors/Agents
T2.2, T4.2 System Inter-comm layer

Figure 3 Security Runtime Monitoring in CPSoSAware: overall approach

Figure 4 provides a more detailed view of the building blocks of the SRMM, distinguishing between architecture components deployed at the system layer, on the upper half of the figure, and components deployed at each individual CPS, on the bottom half of the figure.

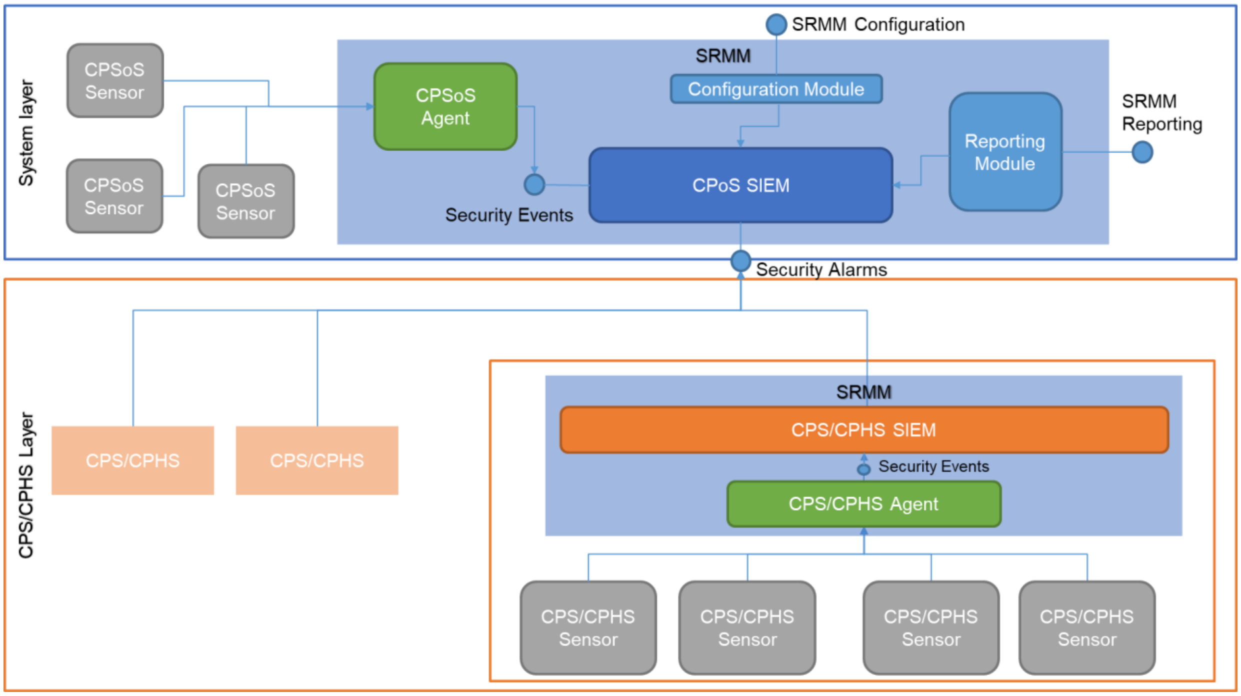


Figure 4 SRMM architecture layers and building blocks

At the system layer, the SRMM is composed of:

- One or more **CPSoS Agents**, in charge of collecting and normalising security information produced by sensors deployed at the system level of the CPSoS. These sensors monitor assets such as the virtual/cloud infrastructure and servers, as well as the inter-communication layer of the system. The CPSoS agents generate as output security events that are pushed to the CPSoS SIEM through the corresponding *Security Events interface*.
- One **CPSoS SIEM**, in charge of the real-time security analysis of all the system-level security events received from the CPSoS Agents and cross-correlation of security alarms produced by the individual CPSs. These alarms are received through the *Security Alarms interface*.
- **Configuration Module** communicates with the CPSoS SIEM to implement the configurations received from other technical components of the CPSoSAware architecture through the SRMM *Configuration interface*.
- **Reporting Module** collects information from the CPSoS SIEM storage system to report results and statistical information to other technical components of the CPSoSAware architecture.

At each CPS, the SRMM is composed by:

- One or more **CPS/CPHS Agents**, in charge of collecting and normalising security information produced by sensors deployed at the CPS. These sensors monitor the assets that compose the CPS, including intra-communication monitoring. CPS Agents generate security events that are pushed to the local CPS/CPHS SIEM through the Security Events interface.
- One **CPS/CPHS SIEM**, in charge of processing security events received from local CPS Agents. Correlation of security events is performed at this level over a limited and very specific set of security directives, customized for threats and anomalies relevant for the specific CPS that is being

monitored. Alarms produced by the CPS SIEM are forwarded to the system-level CPSoS SIEM, for further processing and analysis, but can also be configured to trigger reactions that apply locally to the CPS, without any intervention of the system-level policies.

4 SRMM demonstrator: preliminary version

This chapter describes the preliminary version of the SRMM demonstrator, composed by a SIEM technology, based on the Atos XL-SIEM. After the description of the technology, the SRMM is put into the context of a demonstration scenario in the context of the Autonomous Vehicle use case, to illustrate how the SRMM is deployed, configured and operated, and are the inputs and outputs generated in that particular scenario.

4.1 Technology description: XL-SIEM

Atos Cross-Layer SIEM (XL-SIEM) is a Security Information and Event Management (SIEM) solution deployed on top of the AlienVault's open-source SIEM OSSIM[12], with added high-performance correlation engine to deal with large volumes of security information. It provides scalability and distribution in security events processing through a cluster of nodes, and capacity to raise security alerts from a business perspective based on events collected from different data sources at different layers. These improvements, together with an extended support for data sources, a correlation engine, additional export methods and formats and reaction capabilities provides enhanced features compared to other open-source solutions available in the market. These enhancements, as well as a complete description of the architecture (see Figure 5), functionalities and implementation technologies of the XL-SIEM are detailed in the paper "*Towards an Enhanced Security Data Analytic Platform"*[13]. However, in the following sections and for self-containment of this document, we reuse some of the descriptions from that paper as well as from the online AlienVault OSSIM documentation[14], to briefly explain the main components of the XL-SIEM.

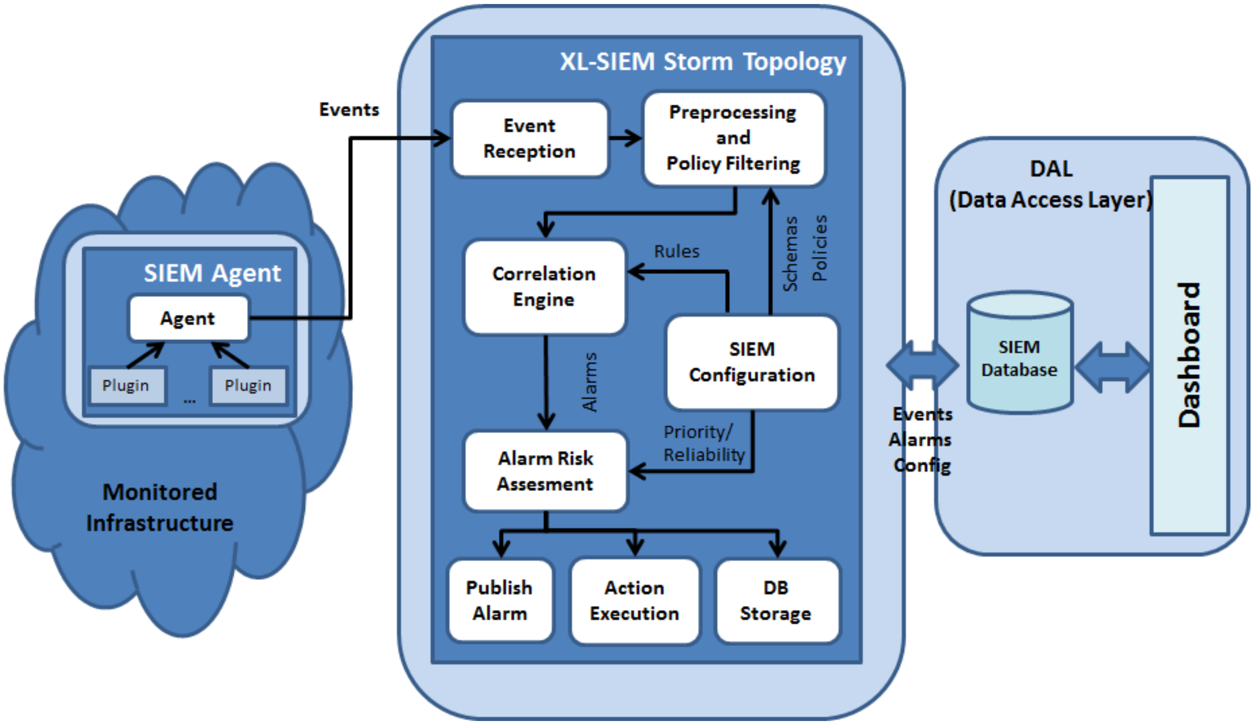


Figure 5 XL-SIEM architecture view [Source:]

4.1.1 Input Data Format

The main input of the XL-SIEM technology is the XL-SIEM Event.

Figure 6 shows the JSON data format of an XL-SIEM Event, based on the original OSSIM Event format[15], which is the output produced by agents.

```
"a": {
        "type": <string>,
        "date": <string>,
        "device": <string>,
        "interface": <string>,
        "plugin_id": <integer>,
        "plugin_sid": <integer>,
        "src_ip": <string>,
        "dst_ip": <string>,
        "src_port": <string>,
        "dst_port": <string>,
        "userdata1": <string>,
        "userdata2": <string>,
        "userdata3": <string>,
        "userdata4": <string>,
        "userdata5": <string>,
        "userdata6": <string>,
        "userdata7": <string>,
        "userdata8": <string>,
        "userdata9": <string>,
        "log": <string>,
        "fdate": <string>,
        "tzone": <string>,
        "event_id": <string>,
        "username": <string>,
        "password": <string>,
        "filename": <string>,
        "organization": <string>
}
```

Figure 6 XL-SIEM Event data: JSON format

Table 2 explains the meaning of each of the fields in the JSON data format.

Table 2 XL-SIEM Event Data: fields description

Fields (*mandatory)	Description

Type*	Type of Agent: monitor, detector
Date*	Date and time of the event (*long* data type)
Device*	IP address of the agent that processed the event
Interface*	Network interface used by the agent
Plugin_ID*	Plugin ID used by the agent to parse and process the raw data
Plugin_SID*	Event type, as defined in the Plugin ID specification
Dst_IP	IP address for the destination of the event
Src_IP	IP address for the source of the event
Dst_Port	Destination port of the event
Src_Port	Source port of the event
Filename	Name of file associated with the event.
Username	The username associated with the event.
Password	The password associated with the event.
Userdata 1-9	User-created fields that can be used freely to log additional information
Organization	The name of the organization that owns the infrastructure where the agent is running
FDate	Full Date and Time the event was logged (ISO 8601 standard format)
TZone	Time Zone
Log	Raw log details that generated the event (Base 64 encoded)
EventID	Unique identifier of the event

Table 3 shows an example of the input and corresponding output produced by the XL-SIEM agent.

Table 3 XL-SIEM agent Input and Output examples

Input to XL-SIEM Agents: syslog
Jul 23 08:58:07 192.168.56.1 [SDRJD][1253]: CRITICAL:sdrjdsyslog:{"jnr": 12.698, "event_duration": 1286, "nodeId": "2", "srcIp": "162.13.144.202", "time": "2018-04-23T14:09:29.000", "freq": "2614000000", "type": "Pulsed", "event": "Attack Running"}
Output from XL-SIEM: XL-SIEM Event
{"event":{"type":"detector","date":"1532329087","device":"212.34.151.202","interface":"eth0","plugin_id":"100000","plugin_sid":"1","src_ip":"162.13.144.202","dst_ip":"212.34.151.202","userdata1":"Mg==","userdata2":"QXR0YWNrIFJ1bm5pbmc=","userdata3":"MjYxNDAwMDAwMA==","log":"SnVsIDIzIDA4OjU4OjA3IDE5Mi4xNjguNTYuMSBbU0RSSkRdWzEyNTNdOiBDUklUSUNBTDpzZHJqZHN5c2xvZzp7ImpuciI6IDEyLjY5OCwgImV2ZW50X2R1cmF0aW9uIjogMTI4NiwgIm5vZGVJZCI6ICIyIiwgInNyY0lwIjogIjE2Mi4xMy4xNDQuMjAyIiwgInRpbWUiOiAiMjAxOC0wNC0yM1QxNDowOToyOS4wMDAiLCAiZnJlcSI6ICIyNjE0MDAwMDAwIiwgInR5cGUiOiAiUHVsc2VkIiwgImV2ZW50IjogIkF0dGFjayBSdW5uaW5nIn0g","fdate":"2018-07-23 06:58:07","tzone":"2.0","event_id":"8e4511e8-9caa-0016-3e3f-a540c116cbee"}}

4.1.2 Agents and Plugins

Plugins are used to instruct agents on how to collect the raw data from security monitoring sensors and how to parse this data to extract the relevant information that will be included in the resulting XL-SIEM events. Plugins are executed by the XL-SIEM Agent which needs to be deployed and configured in a way that it can have access to the data generated by sensors (e.g. log files). For each sensor it is necessary to develop at least one plugin and configure the XL-SIEM agent to use it. Once this is done, the sensor is registered in the XL-SIEM server database as a data source, as it is explained in the OSSIM documentation[16]. This way, the events will be recognized and used appropriately in the XL-SIEM correlation engine. As described in section 2, it is required to develop one plugin for each of the sensors listed in Table 1 that will be used in CPSoSAware Security runtime monitoring infrastructure. Figure 7 shows the internals of an XL-SIEM agent. The figure is just a generalization, showing an agent that is able to parse information collected by all sensors used in CPSoSAware. However, in a real deployment, agents deployed at the system layer will only contain the plugins required to parse the data collected by sensors deployed at that level too. In the same way, agents deployed at each CPS will contain only those plugins required for the sensors deployed at the CPS. This way, resources used by the agents for parsing and event normalization are adjusted to the minimum necessary, to adapt to CPS resource constraints.

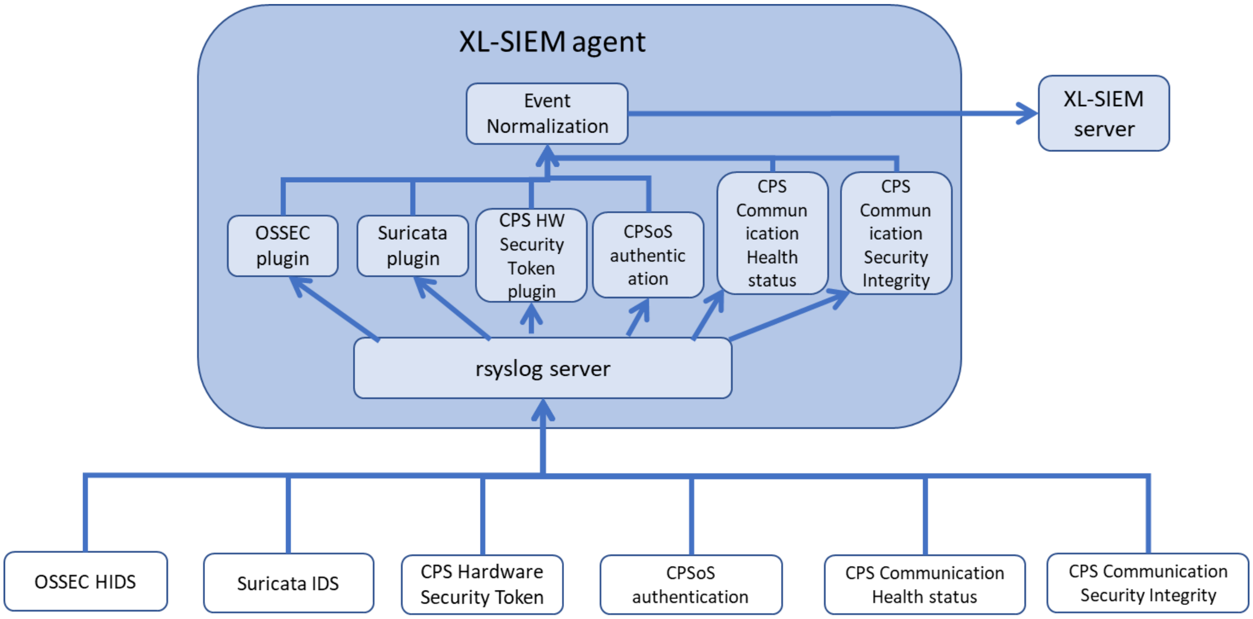

XL-SIEM agent
Event Normalization
XL-SIEM server
OSSEC plugin
Suricata plugin
CPS HW Security Token plugin
CPSoS authentic ation
CPS Commun ication Health status
CPS Commun ication Security Integrity
rsyslog server
OSSEC HIDS
Suricata IDS
CPS Hardware Security Token
CPSoS authentication
CPS Communication Health status
CPS Communication Security Integrity

Figure 7 XL-SIEM agent in CPSoSAware

The XL-SIEM already contains a library of plugins, some are inherited from AlienVault OSSIM SIEM[17] and some others have been developed by Atos from previous deployments in other relevant research projects (such as FINSEC[18], CIPSEC[19] or ANASTACIA[20]) and business cases. To develop new plugins, a corresponding configuration file needs to be created, where among other things it is necessary to define the Regular Expression that parses the original raw data and translates into the corresponding event format fields. Figure 8 shows an excerpt from a plugin that parses the original raw data log produced by a OSSEC HIDS sensor and assigns the relevant information extracted to the appropriate event fields.

```
[0011 - SSH failed loggin attemp]
event_type=event
#precheck="syslog,access_control,authentication_failed"
regexp="^AV - Alert - \"(?P<date>\d+)\" --> RID: "(?P<rule_id>250\d)"; RL: "\d+"; RG:
"(?P<rule_group>syslog,access_control,authentication_failed,)"; RC: "(?P<rule_comment>[^\"]*)"; USER:
\"(?P<username>\S+)\"; SRCIP: \"(?P<srcip>[^\"]*)\";
HOSTNAME:\s\"(?P<agent_name>\([^\)]*\)\s+)?(?:\S+@)?(?P<hostname>(?(agent_name)(?:\d{1,3}.\d{1,3}.\
d{1,3}.\d{1,3})|(?:\S+)))(?:->\S+)?"; LOCATION: \"(?P<location>[^\"]*)\"; EVENT:
"\[INIT\]([^\"]+rhost=(?P<rhost>\S+) user=(?P<ruser>\S+))?([^\"]+)?\[END\]";"
date={normalize_date($date)}
device={resolv($hostname)}
src_ip={resolv($rhost)}
dst_ip={resolv($hostname)}
plugin_sid={$rule_id}
plugin_id={translate($rule_id)}
userdata1={$rule_comment}
userdata2={$rule_group}dst_ip={resolv($local_host)}
dst_port={$local_port}
userdata2={$transport}
```

Figure 8 Excerpt of the OSSEC HIDS plugin configuration file

4.1.3 Events Processing and Security Analysis

Once the events are received from the XL-SIEM agents to the server and before their, the system verifies if the user has specified some conditions to filter the incoming events before they arrive to the correlation engine (e.g., source/destination IP, port, time/date range, type of event, or the SIEM agent where the event is collected). This is done with the definition of Filtering Policies which allow for example to have separated processing and correlation of events from different organisations or realms and comply with legal or business requirements.

The XL-SIEM server improves the existing capabilities of the original OSSIM SIEM with a high-performance correlation engine, implemented with the complex event processing (CEP) Esper[21] (GPLv2 licensed) running in an Apache Storm[22] cluster.

Esper is capable of processing 500,000 events per second with latency below 10 microseconds average with more than 99% predictability. For more complex queries, these values are slightly reduced to a throughput of 120,000 events per second[23], keeping good performance capabilities for processing large volume of data. Security directives or rules are expressed using the Event Processing Language[24] (EPL), which is a declarative programming language that allows expressing security directives with rich event conditions and patterns in a simple way. Table 4 shows an example of a security directive expressed as a correlation rule using EPL language.

Table 4 Correlation rule example

Correlation rule name

BruteForce Microsoft SQL Server authentication attack against SRC IP
EPL Statement
Insert into BruteForce_Microsoft_SQL_Server_authentication_attempt_failed_detected select * from ossimSchema default where (plugin id = 1001) and (plugin sid = 50051 2) and (dst port=1433)
EPL Directive
Insert into directive sls 1 select * from pattern [every-distinct(a.src ip, 15 seconds) a = BruteForce_Microsoft_SQL_Server_authentication_attempt_failed_detected ! ([3] b = BruteForce_Microsoft_SQL_Server_authentication_attempt_failed_detected ((b.src ip=a.src ip) and (b.dst ip=a.dst ip)))]

Apache Storm is a free and open source distributed real-time computation system that working together with Apache Zookeeper and RabbitMQ[25] allows processing the events in a scalable, distributed and fault-tolerant way. A Storm cluster is basically a set of nodes (hosts) where the processing tasks are distributed according to a predefined role. There are two different roles: master and worker. In Storm's terminology, the graph of real-time computation to be executed by the worker nodes is called topology. The latter includes not only the processing logic but also the links indicating how data need to be passed around between nodes. In the topology graph, spouts (sources of streams) and bolts (in charge of data processing) are connected with stream groupings. The XL-SIEM Storm Topology has defined three Spouts to handle input data (i.e. events) from different communication means: RabbitMQ and TCP Socket from agents, plus an additional input via DRPC from the Storm Topology itself. Fifteen Bolts are in charge of the processing, filtering, correlation and cross-correlation, writing to the database, handling internal and external communication and taking actions defined in policies.

This architecture introduces some advantages. Scalability, since the events collected are processed in parallel across a cluster of machines where the parallelism of the different parts of the topology can be scaled individually. Robust process management with the use of Storm and Zookeeper running together. And fault-tolerance, since tasks in a running topology heartbeat to the master node to indicate they are running smoothly. The Nimbus daemon in the master node monitors heartbeats and will reassign tasks that have timed out. Additionally, all the tasks throughout the cluster that were sending messages to the failed tasks quickly will be sent to the new location.

After the processing of the security events in the correlation engine running in the Storm topology, the alarms generated are stored in the database.

4.1.4 Alarms Format, Export and Data Sharing

When one or more events received at the XL-SIEM server match a certain security rule considering different events collected at different layers. These alarms are expressed through a predefined JSON format, and shared with other components, both internal and external, with respect to the organization who hosts the XL-SIEM itself. Figure 9 shows a list of the fields that can be found in the JSON associated to an XL-SIEM alarm:

```
{         "AlarmEvent": {
                    "DST_IP_HOSTNAME": <string>,
                    "RELATED_EVENTS": <string>,
                    "DST_IP": <string>,
                    "PLUGIN_NAME": <string>,
                    "SRC_IP": <string>,
                    "PRIORITY": <integer>,
                    "RELIABILITY": <integer>,
                    "SUBCATEGORY": <string>,
                    "USERDATA3": <string>,
                    "USERDATA4": <string>,
                    "PLUGIN_SID": <string>,
                    "USERDATA1": <string>,
                    "USERDATA2": <string>,
                    "ORGANIZATION": <string>,
                    "CATEGORY": <string>,
                    "PLUGIN_ID": <string>,
                    "USERNAME": <string>,
                    "FILENAME": <string>,
                    "BACKLOG_ID": <string>,
                    "RELATED_EVENTS_INFO": {List of <Event>},
                    "PROTOCOL": <integer>,
                    "RISK": <integer>,
                    "SRC_PORT": <integer>,
                    "SENSOR": <string>,
                    "SRC_IP_HOSTNAME": <string>,
                    "SID_NAME": <string>,
                    "USERDATA7": <string>,
                    "DATE": <string>,  YYYY-mm-dd HH:MM:SS
                    "USERDATA8": <string>,
                    "USERDATA5": <string>,
                    "USERDATA6": <string>,
                    "PASSWORD": <string>,
                    "USERDATA9": <string>,
                    "DST_PORT": <integer>,
                    "EVENT_ID": <string>     }
}
```

Figure 9 XL-SIEM Alarms JSON data format

Some fields are self-explanatory (e.g., EVENT_ID, DATE, SRC_PORT, DST_PORT, DST_IP, SOURCE_IP,…), but others require specific explanation:

- RELATED_EVENTS: XL-SIEM generates alarms considering one or more events who match a certain rule. This field contains the id of the events who led to the generation of the alarm
- PRIORITY: priority value evaluated by the XL-SIEM, associated to the raised alarm
- RELIABILITY: reliability value evaluated by the XL-SIEM, associated to the raised alarm
- RISK: risk value evaluated by the XL-SIEM, associated to the raised alarm. Risk calculation is based on this formula: *Asset Value * Event Reliability * Event Priority / 25 = Risk*
- RELATED_EVENTS_INFO: information about each single event that contributed to the alarm generation. There is an upper limit of the maximum number of events that can be inserted here.
- SID_NAME: high-level description of the alarm
- CATEGORY: category of the alarms
- SUB-CATEGORY: sub-category of the alarm

Besides the native OSSIM data format for Alarms, the XL-SIEM supports export alarms into MISP (Malware Information Sharing Platform)[26] and STIX (Structured Threat Information eXpression)[27] version 2.0 (STIX2), which are standard formats widely used Threat Intelligence Data Sharing.

4.2 Demonstration scenario

For this demonstration scenario we are assuming a set of autonomous vehicles, each one equipped with a set of sensors that improve the driving experience, e.g. in terms of usability, safety; and communication capabilities to connect to other vehicles or smart devices of their environment through V2X technologies. These technologies are very useful and convenient, enabling cooperative information sharing for streamlining traffic movement, improving road safety, etc. But on the other hand, these technologies make autonomous cars vulnerable, and a malicious actor exploiting these vulnerabilities may have serious consequences in the safety of the driver and in the surrounding traffic context. For this reason, it is of paramount importance being able to firstly, monitor and detect anomalies in real-time, secondly correlate these with additional contextual information to determine if there is a security incident happening and thirdly, to assess the risk that this incident may pose to the safety, and potentially to other factors that guarantee the correct and effective functioning and operation of the autonomous vehicle. Being aware of these security incidents in real-time and the risk they pose, permit to warn and to take corrective actions promptly, in order to mitigate their impact and minimise negative consequences.

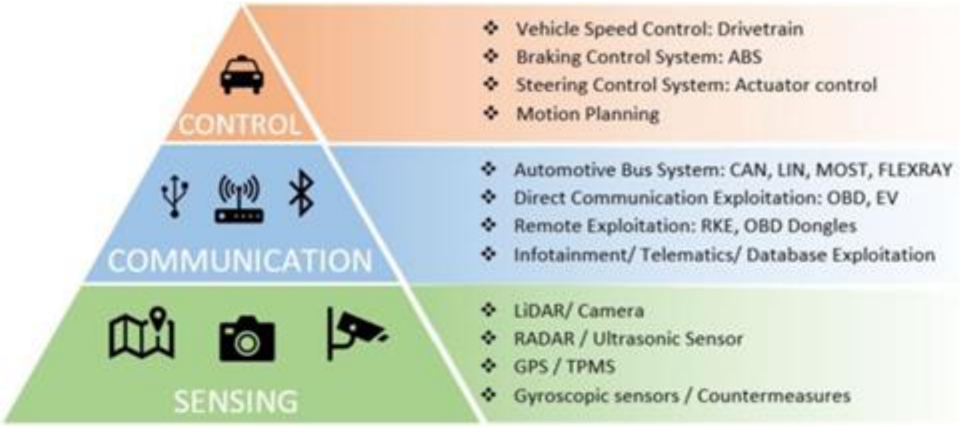

CONTROL
Vehicle Speed Control: Drivetrain
Braking Control System: ABS
Steering Control System: Actuator control
Motion Planning
COMMUNICATION
Automotive Bus System: CAN, LIN, MOST, FLEXRAY
Direct Communication Exploitation: OBD, EV
Remote Exploitation: RKE, OBD Dongles
Infotainment/ Telematics/ Database Exploitation
SENSING
LiDAR/ Camera
RADAR / Ultrasonic Sensor
GPS / TPMS
Gyroscopic sensors / Countermeasures

Figure 10 AV/ADAS vehicle - Security Monitoring Ecosystem [Source: [28]].

Figure 10, which was already included in D6.1 [29], depicts the elements, or assets, that should be monitored and protected in an autonomous car scenario, grouped into three categories: vehicle control module, communication and sensing. This landscape was already introduced in D6.1 but we include it here again for self-containment of this document. We should consider attacks and threats for each of the three categories of assets:

- Threats/Attacks on the sensors. Connected and automated vehicles depend on the collection large volumes of sensors data and processing them to operate safely by maintaining the field of safe travel. Contactless sensors layer attacks can disturb operations of the perception layer of the AV/ADAS stack of the vehicle and can cause hazardous road situations. Practically all types of sensors can be attacked. The following is a non-exhaustive list of attacks that fall under this category:
    - *On-board camera exploit*: this benefits on common vulnerabilities that affect any smart IP camera and that permit attackers to control remotely the device for their own malicious purpose.
    - *GPS sensor spoofing*: A location spoofing attack attempts to deceive a GNSS/RTK receiver by broadcasting incorrect satellite signals, structured to resemble a set of normal satellite signals (e.g., GPS, GLONASS, GALILEO, etc.).
    - *Lidar sensor exploit - Adversarial attacks – data poisoning*: Lidar might be attacked by recording outbound optical signal and sending it back to optical receiver, camera can be disturbed with pointed laser beam (that can also permanently damage its CMOS/CCD sensors) or direct illusional attack on specific classification machine-learning algorithm and ultrasonic sensor can be jammed by generating ultrasonic noise, spoofed by crafted fake ultrasonic echo pulses or even quieted.
    - *Relay attacks – Man-in-the-middle attacks:* can be used for sensor information (e.g. car positioning) stealing, modification and replay.
- Threats/Attacks on the communication, where we can distinguish:
    - Vehicle to Vehicle (V2V) and V2X attacks: with a focus on especially wireless access technologies and attacks detection including diverse types of attacks(e.g. masquerade, wormhole, man-in-the-middle) and assumes three main vectors of attacks for wireless communication: frequency of malicious communication, the effect of the attack on V2X (e.g. injection of fabricated message, message mutation or even preventing delivery of the

message) and effect on the vehicle (e.g. compromising safety or loss of efficiency of the targeted cooperative application).

  - Intravehicular communication attacks, with a focus on CAN bus data manipulation.
  - Distributed Denial of Service (DDoS): consists in attacking a specific service of the system simultaneously and continuously from different sources and using various attacking mechanisms, with the objective of cancelling completely (or significantly degrade) the service. This attack may cause a disruption of the traffic flow, a collision of the vehicle or damages to the infrastructure.

- Threats/Attacks on the vehicle control module: in this category we distinguish side-channel attacks, fault-injection and code-injection attacks, aiming at disclosing, altering and replaying sensible information used by the OBD or the ECU. ECU Firmware tampering or rogue updates have large implications as it can completely reprogram the vehicle's behaviour, resulting in it becoming a potential threat to public safety.

In this deliverable, we are illustrating the use of the SRMM to monitor and detect a threat/attack on the communication layer: a *Distributed Denial of Service (DDoS) attack.* In this type of attack, attackers can be vehicles connected to the network that send a volume of requests higher than what the system can handle, causing a downtime of the service. A *flooding attack* is a specific type of DoS that consist in generating traffic in order to exhaust network resources. In this demonstration scenario, we assume a set of compromised vehicles, controlled by an attacker, that use the flooding attack to disable the wireless network access point. For example, by sending large number of requests for establishing connection, therefore depleting the resources of the node. This will cause other vehicles in the network, who are already attached to it, to be (at least temporarily) isolated from the network. This scenario is graphically depicted in Figure 11.

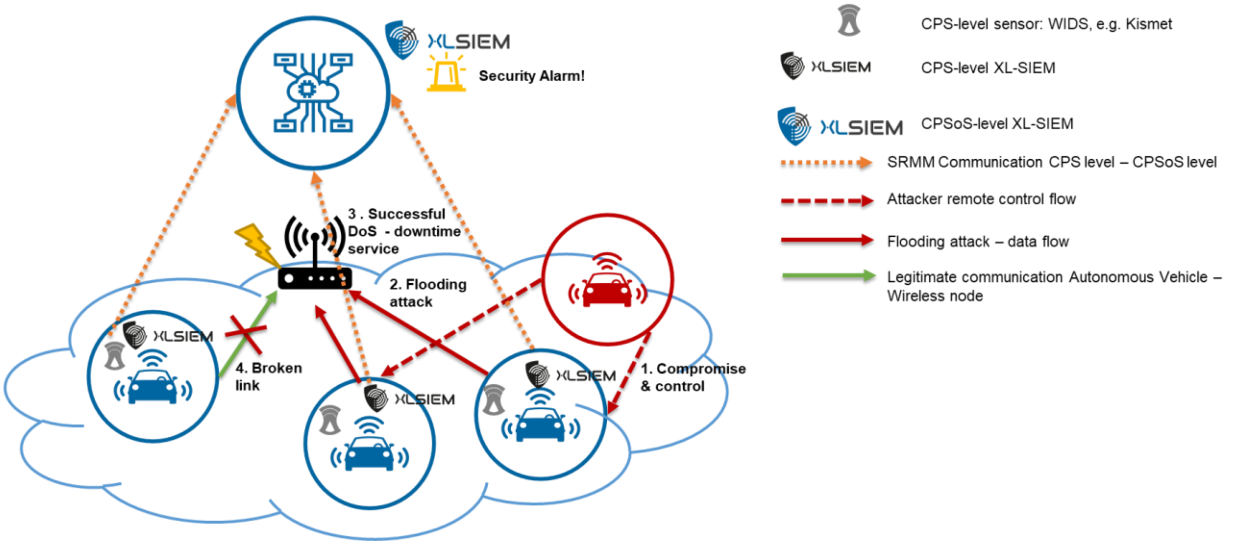

XLSIEM
Security Alarm!
3 . Successful DoS - downtime service
2. Flooding attack
XLSIEM
4. Broken link
XLSIEM
XLSIEM
1. Compromise & control
CPS-level sensor: WIDS, e.g. Kismet
XLSIEM
CPS-level XL-SIEM
XLSIEM
CPSoS-level XL-SIEM
SRMM Communication CPS level – CPSoS level
Attacker remote control flow
Flooding attack – data flow
Legitimate communication Autonomous Vehicle – Wireless node

Figure 11 Demonstration scenario: at CPSoS level

In order to detect this type of attack in a network of autonomous vehicles communicating through wireless, we leverage the capabilities of the SRMM of CPSoSAware in the following way:

- Each autonomous vehicle is equipped with

- a **Wireless Intrusion Detection System (WIDS)**, such as Kismet, acting as a security monitoring sensor
    - a **CPS-level XL-SIEM agent** that collects the security information generated by the WIDS sensor, normalizes it into a security events, and forwards the events to the XL-SIEM server for correlation.
    - A **CPS-level XL-SIEM server** that correlates all the security events received from the agent to alert of an attack that is happening. This instance of the XL-SIEM is a lightweight version that contains a subset of security rules specific to detect threats and attacks relevant for the autonomous vehicle context from the perspective of the individual CPS. Other services such as the XL-SIEM GUI, Apache Storm UI or automatic backups are not included in the CPS-level XL-SIEM. This way, the processing resources required are reduced significantly.
- The system-level SRMM is equipped with **a CPSoS-level XL-SIEM**. This XL-SIEM is a complete version of the technology and contains a full set of security directives relevant for the Autonomous Vehicle context from a wider perspective. For example, security directives that can detect attacks and threats that affect the inter-communication infrastructure, the cloud services, as well as specific rules designed to detect threat scenarios that cross-correlate alarms generated by more than one autonomous vehicle connected to the network and system-level alarms.

Figure 12 shows how the attack is performed at CPS-level.

1- An attacker (depicted in red in the figure) gains access and control to one or more legitimate autonomous vehicles (e.g. through a malware installed or a exploit). A control flow is established between the attacker and the compromised vehicles that permit the attacker to use the legitimate connection between the vehicles and the Wireless AP node to perform malicious actions.

2- The compromised vehicle issues an unusual high number of requests to the Wireless AP (e.g. for disconnecting and re-establishing a connection), on behalf of the attacker.

3- The WIDS sensor running at the vehicle monitors the wireless communication and logs the unusual activity of the vehicle. This is detected by the CPS-level XL-SIEM, through several events collected (De-authenticate/Disassociate flood detected, Unknown de-authentication reason code, Associated Client, New IP detected, Broadcast disassociation detected) and a security alarm warning of a potential data flooding attack is raised at the vehicle. This alarm is also sent to the CPSoS level XL-SIEM for cross-correlation with other alarms raised by this vehicle or others in the realm.

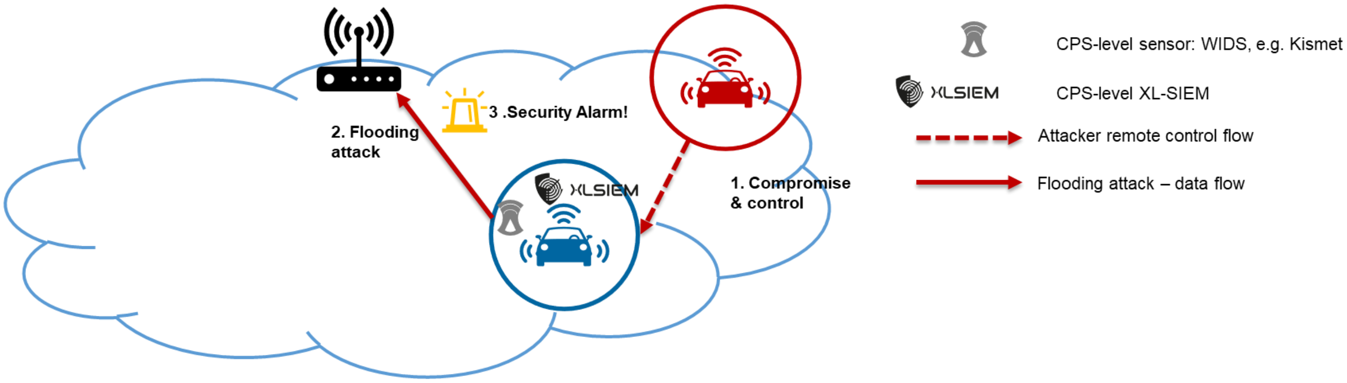

3 .Security Alarm!
2. Flooding attack
XLSIEM
1. Compromise & control
XLSIEM
CPS-level sensor: WIDS, e.g. Kismet
CPS-level XL-SIEM
Attacker remote control flow
Flooding attack – data flow

Figure 12 Demonstration scenario: at CPS level

At the CPSoS level, several alarms of the same type, i.e. “Wireless data flooding attack”, are received from more than one vehicle in the realm (represented by an orange arrow in Figure 11). In the CPSoS XL-SIEM, there is a security directive that is activated when various alarms of type “Wireless data flooding attack” are received from different sources (i.e. different vehicles) and related to the same destination (i.e. Wireless AP) within a small time frame. If this behaviour is observed in the realm, a security alarm is triggered at the CPSoS XL-SIEM indicating that a “Wireless attack, successful denial of service against access point on DST_IP” (where DST_IP is the IP of the Wireless access point).

5 Conclusions and next steps

The Security Runtime Monitoring and Management (SRMM) is a subsystem of CPSoSAware in charge of monitoring the CPSs in order to detect and warn about any suspicious and malicious activity, both at the individual CPS and at the CPSoS levels, which may have an impact in the overall security properties of the system. This document presented the ecosystem that must be monitored, the threats/attacks landscape, and the monitoring, detection and reporting capabilities relevant for that landscape. The document also presents a first version of the CPSoSAware SRMM architecture, its internal building blocks and communication flow, as well as the external interfaces that allow its interaction with other components of the CPSoSAware architecture. The last chapter of the document, content-wise, describes the first version of the SRMM demonstrator, which is based in the XL-SIEM technology of Atos, and which proposes a two-layered deployment of the SRMM in the context of a demonstration scenario based on the Autonomous Vehicle use case. In this demonstrator deployment, each individual CPS is equipped with a lightweight version of the XL-SIEM, with limited resource consumption and specific set of detection capabilities, which communicates with the CPSoS-level XL-SIEM, deployed at the cloud. This two-layered deployment of the SRMM permits distributing the monitoring and detection capabilities at each level. First of all, this approach permits the CPSoS XL-SIEM sharing the workload with the CPSs and this way, avoid bottlenecks at system-level, which will only process dozens of security alerts instead of thousands of security events collected from each vehicle. Secondly, the CPS-level XL-SIEM enables security awareness at the vehicle, permitting the driver or the automatic system to take warn and take decisions based on local security information. Last, the CPSoS XL-SIEM can dedicate only to the security analysis from a general security perspective, considering each vehicle as node of a more complex system, and support taking decisions for mitigation or corrective actions considering the global CPSoS context.

This document has presented an initial version of the technology that implements the CPSoSAware SRMM, which works in isolation with other components and subsystems of CPSoSAware architecture. The next steps can be summarized:

- Identify and characterise new data sources, i.e. the sensors and agents deployed at CPS level, which are defined in T3.5. These need to be integrated with the XL-SIEM and for this, it is required to develop ad-hoc plugins.
- Adapt existing and define new security directives that uses the new data sources and that serve to detect threats and attacks relevant for the Autonomous vehicle scenario.
- Define security policies and actions to respond to triggered security alarms. These can be used to warn/inform, to report to Task 2.1 Data Collection module on a specific format required for metrics calculation, or to initiate mitigation processes or react.
- Design and develop the interfaces that allow configuration and reconfiguration of the SRMM from other components of the CPSoSAware architecture.

This work will be reported in the final version of the SRMM, in deliverable D4.8 which is due in M28.

6 References

[1] Beatriz Gallego-Nicasio Crespo et al., "D1.1 – Supportive, motivating and persuasive approaches, tools & metrics", June 2020.

[2] ENISA Threat Landscape 2020. https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends

[3] ENISA Threat Landscape 2020 - Sectoral/thematic threat analysis. https://www.enisa.europa.eu/publications/sectoral-thematic-threat-analysis

[4] Gartner, "Security Information and Event Management (SIEM) tools Reviews and Ratings". https://www.gartner.com/reviews/market/security-information-event-management

[5] Forrester, "The Forrester Wave™: Security Analytics Platforms, Q4 2020", December 2020.

[6] TechTarget – SearhSecurity, "A guide to SIEM platforms, benefits and features". https://searchsecurity.techtarget.com/buyershandbook/A-guide-to-SIEM-platforms-benefits-and-features

[7] Shostack, A. (2008). Experiences Threat Modeling at Microsoft. MODSEC@ MoDELS, 2008.

[8] Microsoft, "The STRIDE Threat Model", 2009. https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)?redirectedfrom=MSDN

[9] ANASTACIA Project, "D2.4 Secure Software Development Guidelines Initial Report", 2018.

[10] J. Williams, "OWASP Risk Rating Methodology", https://owasp.org/www-community/OWASP_Risk_Rating_Methodology

[11] Pavlos Kosmides, Eleni Adamopoulou et al.. "D1.3 – Preliminary Version of CPSoSaware System Architecture". December 2021.

[12] https://www.alienvault.com/

[13] Gustavo Gonzalez-Granadillo, Susana Gonzalez-Zarzosa and Mario Faiella, "Towards an Enhanced Security Data Analytic Platform". 15th International Conference on Security and Cryptography (SECRYPT), 2018.

[14] https://www.alienvault.com/documentation/resources/pdf/usm-appliance-user-guide.pdf

[15] https://www.alienvault.com/documentation/usm-appliance/events/event-details-fields.htm

[16] https://www.alienvault.com/documentation/usm-appliance/plugin-management/tutorial-developing-new-plugin.htm

[17] https://www.alienvault.com/docs/data-sheets/usm-plugins-list.pdf

[18] FINSEC Project. www.finsec-project.eu

[19] Cipsec Project. www.cipsec.eu

[20] ANASTACIA Project. www.anastacia-h2020.eu

[21] 5http://www.espertech.com/esper/

[22] http://storm.apache.org/

[23] Mathew, A. (2014). Benchmarking of complex event processing engine esper. Technical Report

[24] https://docs.oracle.com/cd/E13157 01/wlevs/docs30/epl guide/overview.html

[25] https://www.rabbitmq.com/

[26] http://www.misp-project.org

[27] Barnum, S. (2014). Standardizing cyber threat intelligence information with the structure threat information expression

(stix). Whitepaper.

[28] Zeinab El-Rewini, Karthikeyan Sadatsharan, Daisy Flora Selvaraj, Siby Jose Plathottam, Prakash Ranganathan,
“Cybersecurity challenges in vehicular communications”, Vehicular Communications, Volume 23, 2020, 100214,
ISSN 2214-2096, https://doi.org/10.1016/j.vehcom.2019.100214.

[29] Michał Niezgoda et al., “D6.1 – Definition and Planning of Quantification Trials”, 2020.