D4.8 – FINAL SECURITY THREAT AND ATTACK MODELS

Deliverable ID	**D4.8**
Deliverable Title	**Final Security Threat and Attack Models**
Work Package	**WP4**
Dissemination Level	**PUBLIC**
Version	**1.0**
Date	**30/11/2019**
Status	**final**
Lead Editor	**Ákos Milánkovich (SLAB)**
Main Contributors	**Arthur Pitman (UNI-KLU)** **Etienne Brosse (SOFTEAM)**

Published by the CPSwarm Consortium

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 731946.

Document History

Version	Date	Author(s)	Description
0.1	2018-10-29	Regina Krisztina Bíró (SLAB)	First Draft with TOC
0.2	2019-09-09	Ákos Milánkovich (SLAB)	Second draft with previous content and contribution requests
0.3	2019-09-27	Arthur Pitman (UNI-KLU)	Integration of safety summary
0.4	2019-10-31	Ákos Milánkovich (SLAB)	Added use-case specific content, SOFTEAM contribution
1.0	2019-11-30	Ákos Milánkovich (SLAB)	Integrated the suggestion from reviewers.

Internal Review History

Review Date	Reviewer	Summary of Comments
2019-11-04	Davide Conzon (LINKS)	Approved with minor comments.
2019-11-04	René Reiners (FIT)	Approved with minor comments.

Table of Contents

1 Introduction

1.1 Goal

In order to help the users of the CPSwarm Workbench build secure products, this document endeavours to describe the threat landscape autonomous swarms face, with two main goals:

- To provide guidance for developers on security aspects when using the CPSwarm Workbench,
- To aid the development of a workbench that acts as an enabler for security and safety features.

The reader of this document, by the end, should have a solid, high-level understanding of the threats faced by an autonomous swarm.

This deliverable is an extension of D4.7 Initial security threat and attack models, which introduced general aspects of security assessment for the CPSwarm project. This deliverable aims to extend the findings with use-case-specific details.

1.2 Summary

In Chapter 2, the authors provide an overview of the different methodologies that are used today in the field of safety and security to describe threats and risks in a systematic way. The rationale for selecting and adjusting CPSwarm methodology is also described.

In Chapter 0, relevant regulations and standards are described. This is expanded with use-case specific case studies that helped us – and will help others – understand the threats swarms face.

In Chapter 4, the case studies of the project are introduced. Their assessment is handled in the following chapters.

Chapter 5 describes the safety aspects for the project and the use-cases.

In Chapter 6, the authors identify and categorize the assets that are relevant to an autonomous swarm and also investigate the security relevant properties of each asset using the Confidentiality Integrity Availability (CIA) triad.

In Chapter 7, attacker motivations and attack methods will be explored using attack trees, in order to understand the kind of attacks operators of swarms may face and how these attacks affect the assets previously identified.

In Chapter 8, the authors categorize the low-level attacks that have been discovered in order to define the scope of the countermeasures that need to be developed and to propose concrete countermeasures.

In Chapter 9, the authors assess the risk of each of these attacks, based on their likelihood and severity – both with and without the proposed countermeasures.

In Chapter 10, the authors recap the contents of the deliverable and concludes it.

1.1 Related documents

Title	Reference	Version	Date
Final Vision Scenarios and Use Case Definition	D2.2	1.0	M16
Initial Automotive Demonstration	D8.5	1.0	M24
Initial Swarm Logistics Demonstration	D8.3	0.2	M24
Initial Swarm of Drones and Ground Robots Demonstration	D8.1	0.2	M24
Initial Security Threat and Attack Models	D4.7	1.0	M22

Final CPSwarm Abstraction Library	D7.2	-	M34
Final Validation Results	D8.8	-	M36
Final CPSwarm Abstraction Library	D7.2	-	M32
Final Bulk Deployment Tool	D7.4	1.3	M32
Final Modeling Library	D4.3	-	M33
Final Lessions Learned and Requirements report	D2,7	0.6	M26

2 Methodology

Usually as one of the first steps of conducting a security analysis, threat modelling takes a look at the system from the attackers' perspective. Its goal is to identify high value assets, as well as potential vulnerabilities and threats. Over time, a number of methodologies have been developed to establish a systematic way to conduct security analyses, the most famous of which is STRIDE/DREAD, the words themselves being mnemonics for their respective categories. Using these methodologies, threats can be identified, risks can be assessed and decisions can be made on the development of countermeasures.

2.1 State of the art research on methodologies

This chapter summarizes the most widely used threat modelling methodologies both in the security and safety domain. Concerning security, the following methodologies are going to be discussed: STRIDE, DREAD, Trike, P.A.S.T.A. and VAST; as well as Attack Trees as a method to assist in the description of threats. For safety, the terminology is fault modelling as most threats to safety usually arise from component failure – here FMEA and FTA are going to be discussed.

2.1.1 STRIDE

The STRIDE [1] methodology aims to classify known threats according to the kinds of exploits used or motivation of the attacker. It was one of the first threat modelling approaches developed and published by Microsoft in 1999. The name, STRIDE is an acronym formed from the first letter of each of the following categories used by it for threat classification:

- **Spoofing identity:** "Identity spoofing" is a key risk for applications that have many users but provide a single execution context at the application and database level. In particular, users should not be able to become any other user or assume the attributes of another user.
- **Tampering with data:** Users can potentially change data delivered to them, return it, and thereby potentially manipulate client-side validation, GET and POST results, cookies, HyperText Transfer Protocol (HTTP) headers, and so forth. The application/product in question should not send data to the user, which are obtainable only from and/or within the application/product itself. It should also carefully check data received from the user and validate that it is valid and applicable before storing or using it.
- **Repudiation:** Users may dispute transactions if there is insufficient auditing or recordkeeping of their activity. For example, if a user says, "But I didn't transfer any money to this external account!", and it is not possible to track his/her activities through the application, then it is extremely likely that the transaction will have to be written off as a loss. This is usually tackled by non-repudiation controls, such as web access logs, audit trails, etc.
- **Information disclosure:** Users are rightfully wary of submitting private details to a system. If it is possible for an attacker to publicly reveal user data at large, whether anonymously or as an authorized user, there will be an immediate loss of confidence and a substantial period of reputation loss. Therefore, it is a must to include strong controls to prevent user ID tampering and abuse, particularly if a single context is used to run the entire application/product.
- **Denial of service:** Designers should be aware that their applications/products may be subject to a denial of service attack. Therefore, the use of expensive resources such as large files, complex calculations, heavy-duty searches, or long queries should be reserved for authenticated and authorized users, and not available to anonymous users.
- **Elevation of privilege:** If an application/product provides distinct user and administrative roles, then it is vital to ensure that the user cannot elevate his/her role to a higher privilege one. In particular, simply not displaying privileged role links is insufficient. Instead, all actions should be gated through an authorization matrix, to ensure that only the permitted roles can access privileged functionality.

2.1.2 DREAD

DREAD [2] is another classification scheme designed by Microsoft used to quantify, compare and prioritize the amount of risk induced by each evaluated threat. The name of the methodology is an acronym formed from the first letter of the following categories to evaluate the risk level of a given threat:

- Damage potential
- Reproducibility
- Exploitability
- Affected users
- Discoverability

The DREAD risk level is calculated by adding the scores from each of the above categories and the dividing it by 5 – to obtain an average. The scores range from 0 to 10 – 0 is the lowest meaning minimal or no impact induced by the threat and that it is extremely difficult to perform and scale; while 10 is the highest score and indicates that the threat is easy to perform/scale or has a severe/widespread impact.

2.1.3 Trike

The Trike [3] methodology focuses on using threat models as an input for risk management. The risk based approach Trike is mostly used by auditing teams where the threat models are based on a "requirements model". This establishes an acceptable level of risk defined by stakeholders attached to each asset class. Threats build upon these requirements and are then assigned a risk value.

2.1.4 P.A.S.T.A.

The Process for Attack Simulation and Threat Analysis (PASTA) [4] is a seven-step, risk-centric methodology. It aligns business objectives and technical requirements by providing a seven-step process which takes compliance issues and business analysis into account. The seven steps of the methodology are the following:

1. **Define objectives:** business objectives, security and compliance requirements
2. **Define technical scope:** define the boundaries of the technical environment and the underlying infrastructure, applications and software dependencies
3. **Application decomposition:** identify use cases, define application entry points and trust levels, identify actors, assets, services, roles and data sources
4. **Threat analysis:** probabilistic attack scenarios analysis, regression analysis on security events and threat intelligence correlation and analytics
5. **Vulnerability and weaknesses analysis:** queries of existing vulnerability reports, issues tracking, mapping threats to existing vulnerabilities, design flaw analysis using use and abuse cases, scorings (Common Vulnerability/Weakness Scoring System and Common Vulnerability/Weakness Enumerations.)
6. **Attack modelling:** attack surface analysis, attack tree development and attack library management, attack to vulnerability and exploit analysis using attack trees
7. **Risk and impact analysis:** qualify and quantify business impact, countermeasure identification and residual risk analysis, risk mitigation strategies

2.1.5 VAST

The unique underlying principle of the Visual, Agile and Simple Threat modelling (VAST) [5] methodology is the scalability of the threat modelling process throughout the infrastructure and software development lifecycle, integrating threat modelling into an agile software development methodology.

2.1.6 Attack trees

The attack tree approach [6] identifies possible threats with conceptual diagrams called attack trees consisting of one root node, internal nodes, and leaf nodes. From the bottom up, nodes are conditions, which must be satisfied in order to make the direct parent node true. There are two types of parent nodes: *AND* and *OR* (see Figure 5). In the first case, every child node must be satisfied, while in the second case, one is enough. When the root node of an attack tree is satisfied, the attack is complete.

An example for attack trees is provided by the H2020 COSSIM project [7], which aimed to provide an integrated CPS simulation framework. Figure 1 depicts an attack tree describing an attacker's objective to obtain confidential data sent between the COSSIM Framework elements.

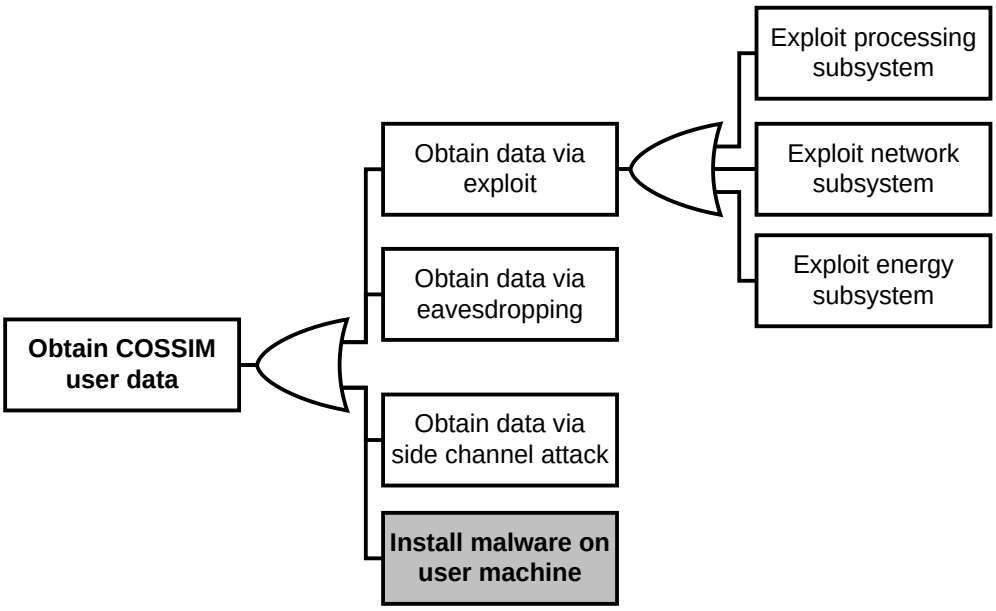

Figure 1 - Attack tree describing an attacker obtaining user data, from the COSSIM project [7]

The internal and child nodes represent sufficient attacks to reach the root node representing the goal of the adversary. It suggests that an attacker may be able to

1. obtain user data by exploiting a vulnerability in the COSSIM processing subsystem's implementation and read the data via direct memory access
2. perform eavesdropping on the communication between different components, and obtain data in that way as well
3. perform a side channel attack against the processing subsystem or energy subsystem
4. install malware on a COSSIM user's computer and steal confidential data from the user directly.

2.1.7 Failure mode and effects analysis (FMEA)

As a structured and systematic technique for failure analysis, FMEA [8] has been in use for over 50 years to assess the reliability and safety of critical systems. Its main goal is to identify failure and eliminate (or at least minimize) the number of catastrophic failure conditions. It is an inductive process, as it considers a single failure at a time and examines its effect on the system as a whole. The analysis aims to identify and eliminate all single points of failure for the system, and should be conducted in parallel with the design process, to minimize the cost of developing countermeasures (see Figure 2).

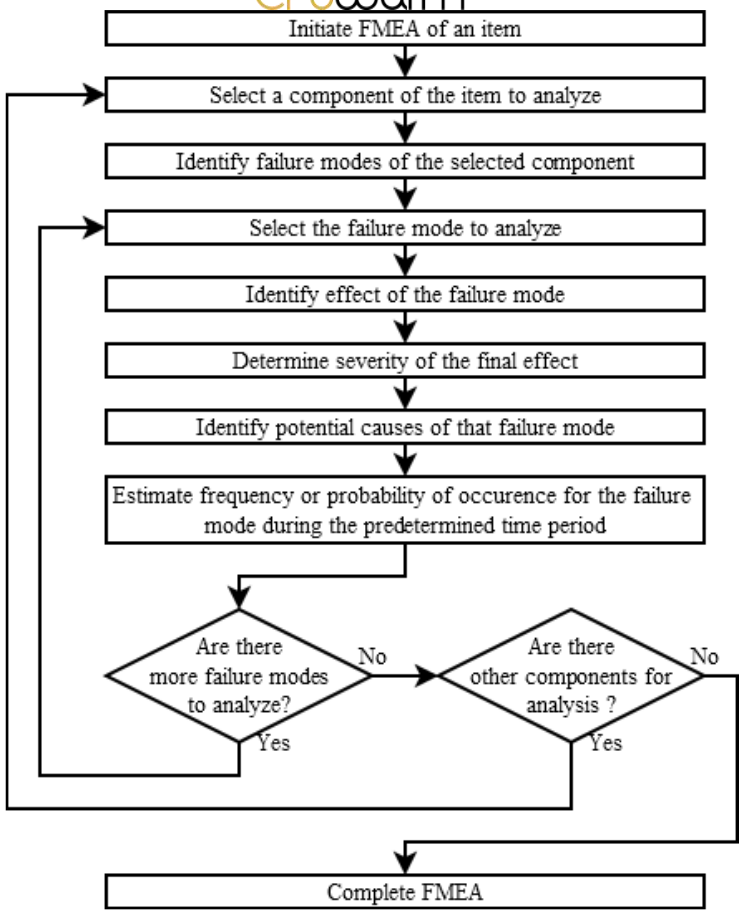

Figure 2 - FMEA flowchart [9]

2.1.8 Fault tree analysis (FTA)

Similar to attack trees, FTA [10] builds an event tree for system failures (see Figure 3) using Boolean logic to combine fault indications from system components. Deductive reasoning is used to determine how different events contribute to a single system failure condition – starting from the top, the system state that needs to be avoided, and working backwards, trying to establish when that condition can occur. When events on which such a tree is built are combined with their probabilities, the tree itself can be used to ascertain the probability of system failure and to identify areas in need of additional countermeasures. A welcome side-effect of the analysis is that the resulting graph can also be used as a service manual for identifying the root cause of system problems.

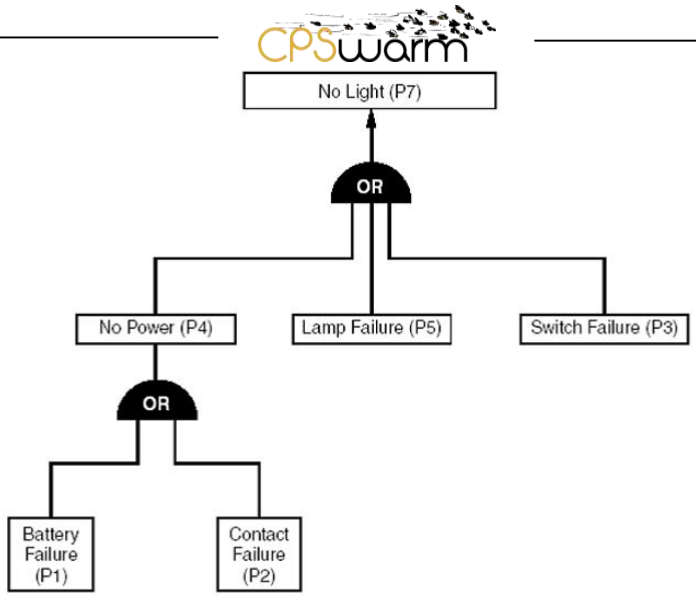

Figure 3 - Fault tree analysis on a vehicle headlamp [11]

2.2 Establishing CPSwarm methodology

Since the CPSwarm project deals with CPSs which are connected with the physical world, it is desirable to be able to model both security and safety threats and faults. This is why Attack Trees and Fault Trees were chosen as tools for security and safety modelling. To get a unified solution – since the syntax and design elements are the same in both cases – the CPSwarm Consortium has developed a plugin for Modelio for attack and fault tree analysis. This will ease the user workflow concerning safety and security as users can link their findings during the threat and fault analysis to existing security and safety solutions in the D4.3 Final Modeling Library.

The threat modelling methodology, on the other hand, would not be taken directly from the above listed techniques – as most of these methodologies are designed for application security. However, since the CPSwarm project has three use cases and different stakeholder needs, this deliverable will follow the outline of P.A.S.T.A. – starting with establishing the scope of the work using a case study in the following chapter. Chapter 5 deals with defining generic and use-case specific assets for all use cases, Chapter 6, 8 and 9 will address the traditional threat models, risk assessment and countermeasures catalogue. These chapters can be considered as security evaluation guidelines and examples for CPS.

2.3 Evaluation guidelines

This subsection provides detailed guidelines on evaluating devices and software used by swarms from a security perspective – security relevant validation activities required by the Validation Framework and performed during Use Case Validation rely on these guidelines.

For commercial deployment of swarms in real, production environments a high level of security assurance is required. While no legal framework exists as of now for most application areas that govern the security requirements for CPS, work is underway both in European Union (EU) organizations and worldwide on developing a common set of requirements and criteria that can be used when evaluating IoT devices. As a contribution to this effort and as an extension of the existing methodology used by Search-Lab (see Meforma [12]) and as part of an ongoing effort to develop a commercial certification scheme that can be used to describe the security level of these devices.

2.3.1 MEFORMA overview

MEFORMA is a security evaluation methodology designed to be customer-oriented, meaning that the evaluations are being accomplished on a project basis using up resources fixed in advance, and the outcomes not only provide a passed-or-failed result like most of the certification schemes, but by the

recommendations given, the development groups also receive valuable support on how to correct the found problems.

Aligned to the usual terminology, Target of Evaluation (ToE) denotes the system being evaluated, and we have two simple roles, the Developer and the Evaluator.

The project approach implies that the work is accomplished in different phases that build upon each other, and that each phase ends with providing a deliverable documenting the results. A typical MEFORMA evaluation project consists of the following phases:

- **Preparation Phase:** The test environment is established, and threat modelling is performed on the ToE – based on its results, test cases are specified. Deliverable is the ***Evaluation Plan*** that contains the definition of the scope, the identified security objectives, the threat model, and the test cases.
- **Evaluation Phase:** The defined test cases are executed, confirming whether the originally identified threats are viable or not. Verified threats (findings) are reported to the Developer regularly through ***Weekly Status Reports*** documenting the progress of the evaluation.
- **Documentation Phase:** The findings are collected, all threats are enlisted and a risk analysis is performed. Most importantly, recommendations are given to the Developer explaining how it should deal with each threat. All results are compiled in the ***Evaluation Report*** as the main deliverable of the project.
- **Review Phase:** During this final phase, a new and fixed version of the ToE is re-evaluated (regression testing) to determine whether the identified threats had been adequately addressed. Evaluation Report is updated with the new results, forming the ***Review Report***.

2.3.2 Evaluation process

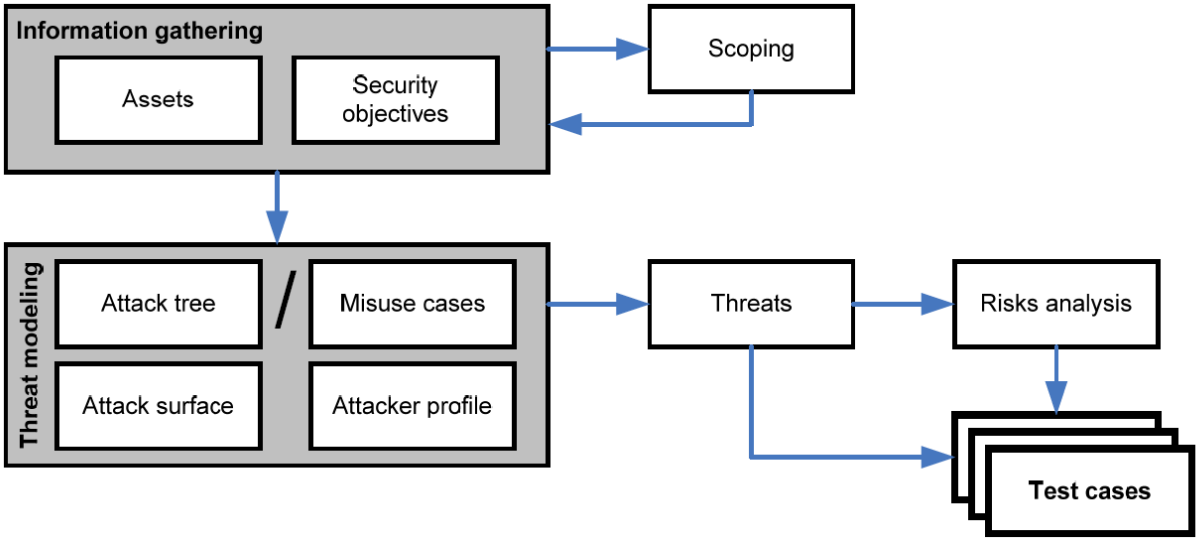

Figure 4 - MEFORMA

As in Figure 4 the first step is to identify the ToE, and the scope of the evaluation must be specified, which is a co-operative effort between the Evaluator and the Developer. Basically, there are three main aspects of planning an audit: scope, depth of analysis and the audit risk. The main goal of this step is to specify the security objectives. For this, one should first identify and understand the assets (e.g., Chapter 4 and 5) within the system that need to be protected, and then for each asset determine which of the independent security objectives (typically taken from the CIA triad i.e. Confidentiality, Integrity, and Availability) are relevant. Results of Scoping, Information Gathering, Threat Modelling (Chapter 4 to 9) and Test Case Specification are all summarized in the Test Plan, which is refined and agreed with the Developer through several iterations. Building upon good preparatory work, the evaluation simply means the execution of the test cases already specified. Actual evaluation of a test case can consist of black-box / white-box / grey-box testing or source code review, and can also include reverse-engineering of the system.

2.4 Threat modelling using Modelio

As planned previously mentioned in D4.7, a Modelio module (aka extension) has been developed to help users of the CPSwarm Workbench to perform security modelling within the same ecosystem. This not only makes it easier for engineers working with the Workbench to systematically describe threats, the mere presence of the feature prompts users to explore their own threat landscape and include the results of their findings in their development work.

Attack trees have been proved to be useful in threat analysis due to their simple and unambiguous concepts. Attack Trees were introduced by [13] as a formal way of describing the security of systems, based on a variety of attacks. Basically, attacks against a system are represented in a tree structure, with the goal as the root node and different ways of achieving that goal as leaf nodes. A node is connected to its children with one of the 2 types of conditions: AND and OR conditions. The Figure 5 shows examples of attack trees where the goal described here as Attack "A" requires the realization of both Attacks "B" **AND** "C".

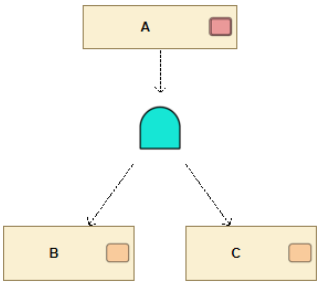

Figure 5 - Example of an attack tree with an AND clause

The Figure 6 shows a diagram where the Attack "A" can be realized by either realizing Attack "B" **OR** "C".

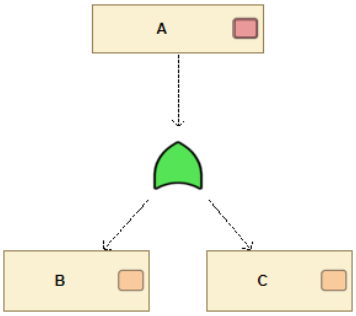

Figure 6 - Example of an attack tree with an OR clause

2.5 Security requirements

The following project-specific security and safety-related requirements were formulated in the planning phase of the project (as presented in D2.7 Final Lessions Learned and Requirements report). The fulfilment of the requirements is summarized in Chapter 8.3.

ID	Description
CRD-143	Passwords shall never be viewable at the point of entry or at any other time.
CRD-133	The system shall not be shut down for maintenance more than once in a 24-hour period.
CRD-128	The system shall be protected against cyber attacks
CRD-127	Attempts at accessing sensitive data by unauthorised users must be logged

ID	Description
CRD-126	Accessing sensitive data must be logged (User ID, Timestamp, etc.)
CRD-123	The solution should be in compliance with GDPR as well as national policies
CRD-119	Data processing and management must comply with relevant regulations
CRD-81	Software components running on the CPS shall be started with the lowest possible privileges.
CRD-78	The Deployment Agent shall use the list of trusted certificates supplied when the device is first provisioned to validate signatures.
CRD-76	The Deployment Manager shall provide a way to generate, import and export operator specific keys for code signatures.
CRD-75	The Deployment Agent shall verify the signatures of packages on boot and when updates are received.
CRD-73	The Deployment Tool shall implement secure over-the-air update functionality.
CRD-72	The Deployment Manager shall sign all packages with an operator specific key.
CRD-68	All communications between swarm members shall be authenticated and integrity protected, with a per-message policy on encryption.
CRD-67	All communications between the swarm and the tools in the workbench shall be authenticated, integrity protected and encrypted.
CRD-64	The Code Generator and all the code generated shall be compliant to ISO 26262.
CRD-60	The communication between the Deployment Agent running on swarm members and the Deployment Manager shall be authenticated, authorized, encrypted, and integrity checked.
CRD-35	The communication link between the swarm and the Monitoring Tool shall be authenticated and encrypted

3 Regulations and industry standards

3.1.1 Regulations

As of now, the EU does not have specific legislation on robotics. Robots as products are regulated by a number of legislative frameworks, such as the Directive on Liability for Defective Products [14] and the Product Safety Directive [15]. To review the regulatory challenges posed by the advancing robotics technology, the Robolaw [16] project was funded under the European Commission's 7th Framework Programme for Research and Technological Development (FP7). The main objective of the Robolaw project was assessing whether existing regulations in the EU are sufficient to address new problems brought by robotics technology and ensuring that the regulations provide conditions which incentivize European innovation in the robotics sector. Since the current and future regulations of robotics are complicated by the fact that there is no common understanding of what a robot is, the Robolaw project addressed this problem by identifying four categories where the application of the existing EU legislation would be problematic. These categories are

- Driverless vehicles
- Robotic prostheses
- Surgical robots
- Robot companions

Comparing differences and similarities of the above four categories, the Robolaw project proposed five main features with which robots can be categorized: autonomy, human-robot interaction, nature, environment and task. Based on these five features, the European Parliament has agreed on the characteristics that describe "smart robots":

1. Acquisition of autonomy through collecting data through sensors or exchanging data with its environment and analysing the data
2. Self-learning from experience or interaction
3. Having physical hardware components
4. Being able to adapt its behaviour and actions to dynamic environments
5. The absence of life in the biological sense

In the near future, the European Commission intends to analyse the above criteria and decide whether it is necessary for future regulatory purposes. Moreover, the Commission is planning to create definitions for three main categories within "smart robots", namely Cyber-Physical Systems, Autonomous Systems and Smart Autonomous Robots.

Since robotics technology is still a fairly new branch of industry, there are a number of upcoming regulatory and policy initiatives which are expected to be implemented by the European Commission:

- **Civil Law Liability:** the Commission will address legal questions related to the development and use of robotics and artificial intelligence in the next decade, and has already launched an evaluation of the Directive on Liability for Defective Products [17]. The extent to which the Directive can be applied to new technological developments, including advanced robotics and autonomous systems still needs to be evaluated.
- **Product Safety:** The Machinery Directive is currently being evaluated by the Commission to add better regulation principles. The revision may adapt the Directive's health and safety requirements to autonomous robots.
- **Autonomous cars and testing:** The Commission has launched several initiatives concerning autonomous cars, such as the European strategy on Cooperative Intelligent Transport Systems,

connected and automated mobility (C-TIS) [18] and intends to establish cross-border testing corridors for these systems.

- **Harmonization of technical standards:** The are a number of research activities addressing the development of testing protocols for cooperative and collaborative systems which may lead to the creation of safety certification standards specific to the robots subject to these research projects.
- **An Advisory Body for Robotics and Artificial Intelligence:** the Commission proposes to create a high-level advisory body on robotics to advise the Commission.

Although there is a lot remaining to be decided, it is clearly foreseeable that the Commission's actions will significantly affect the development and research of robotics and artificial intelligence in the EU.

3.1.2 Industry standards

This subsection deals with the introduction of currently available security and safety standards in the robotics industry. ISO 12100[1], Safety of machinery - General principles for design - Risk assessment and risk reduction

The ISO 12100 standard specifies the basic terminology, principles and methodology for achieving safety in machinery design. It specifies risk assessment and risk reduction principles to help designers achieve their objectives. The standard is intended to be used as a basis for the preparation of

- type-B (generic) safety standards, which deal with one safety aspect or one type of safeguard that can be used across a wide range of machinery and
- type-C (machine) safety standards dealing with detailed safety requirements for a particular machine or group of machines.

3.1.2.1 ISO 13849-1, Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design

The ISO 13849-1[2] standard introduces required performance levels for safety-related control systems. The performance levels can be applied to the following safety-related parts of control systems:

- protective devices (e.g. two-hand control devices, interlocking devices), electro-sensitive protective devices (e.g. photoelectric barriers)
- control units (e.g. a logic unit for control functions, data processing, monitoring, etc.)
- power control elements (e.g. relays, valves, etc.)

3.1.2.2 IEC 62061, Safety of machinery - Functional safety of safety-related electrical, electronic and programmable electronic control systems

The IEC 62061[3] standard defines safety requirements for hardware and software and assigns safety integrity levels (SIL) for safety-related control systems (SRECS) as seen in Figure 7.

[1] https://www.iso.org/standard/51528.html
[2] https://www.iso.org/standard/69883.html
[3] https://webstore.iec.ch/publication/6426

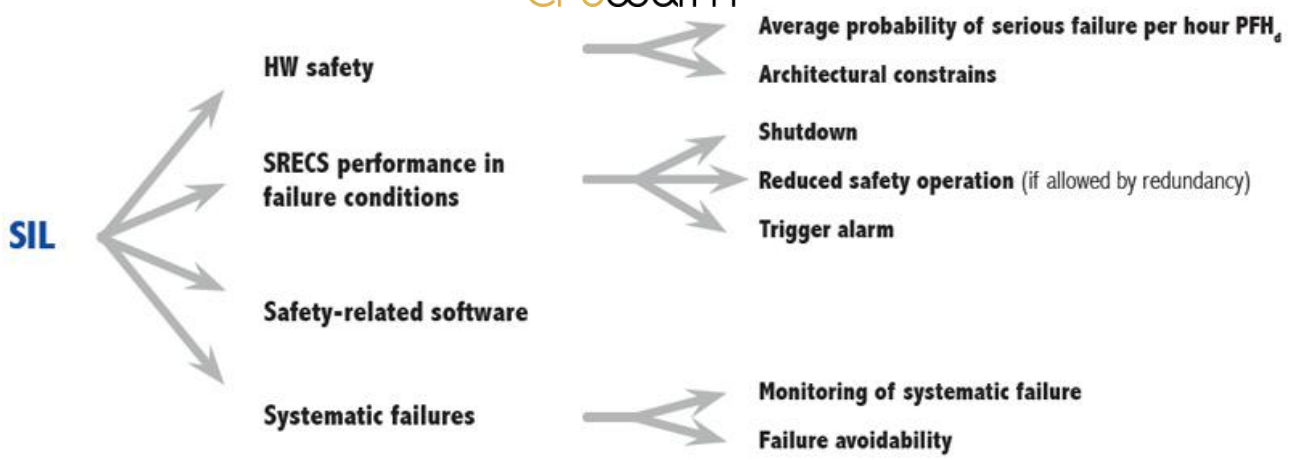

Figure 7 - Safety Integrity Levels [19]

3.1.2.3 ISO 10218-1, Robots and robotic devices - Safety requirements for industrial robots -Part 1: Robots Requirements for the design of manipulators for industrial environments

The ISO 10218-1[4] standard specifies requirements and guidelines for the inherent safe design, protective measures and information for the use of industrial robots. It describes basic hazards associated with robots and provides requirements to eliminate, or adequately reduce, the risks associated with these hazards.

3.1.2.4 ISO 10218-2, Robots and robotic devices – Safety requirements for industrial robots – Part 2: Robot systems and integration

The ISO 10218-2[5] standard specifies safety requirements for the integration of industrial robots and industrial robot systems as defined in ISO 10218-1, and industrial robot cell(s). The integration includes the following:

- the design, manufacturing, installation, operation, maintenance and decommissioning of the industrial robot system or cell;
- necessary information for the design, manufacturing, installation, operation, maintenance and decommissioning of the industrial robot system or cell;
- component devices of the industrial robot system or cell.

Some examples include collaborative modes like monitored stop, hand guiding, velocity and/or force control.

3.1.2.5 ISO/TS 15066, Robots and robotic devices – Collaborative robots

The ISO/TS 15066[6] standard provides guidance for collaborative robot operation where a robot system and people share the same workspace. To achieve safety, robotic applications traditionally exclude operator access to the operations area while the robot is active. Therefore, a variety of operations requiring human intervention often cannot be automated using robot systems. In such operations, the integrity of the safety-related control system is of major importance, particularly when process parameters such as speed and force are being controlled.

[4] https://www.iso.org/standard/51330.html
[5] https://www.iso.org/standard/41571.html
[6] https://www.iso.org/standard/62996.html

3.1.2.6 ISO 13482, Robots and robotic devices – Safety requirements for personal care robots

The ISO 13482[7] standard specifies requirements and guidelines for the inherently safe design, protective measures, and information for use of personal care robots, in particular the following three types of personal care robots:

- Mobile servant robots
- Physical assistant robots
- Person carrier robots

3.1.2.7 ISO 13850 Specification of functional requirements and design principles

The ISO 13850[8] Standard specifies functional requirements and design principles for the emergency stop function on machinery, independent of the type of energy used.

It does not deal with functions such as reversal or limitation of motion, deflection of emissions (e.g. radiation, fluids), shielding, braking or disconnecting, which can be part of the emergency stop function.

3.1.2.8 ISO/IEC 15408, Common Criteria for Information Technology Security Evaluation Threat and Attack Models in Swarms of Cyber-Physical Systems

ISO/IEC 15408-1:2009[9] establishes the general concepts and principles of Information Technology (IT) security evaluation and specifies the general model of evaluation given by various parts of ISO/IEC 15408 which in its entirety is meant to be used as the basis for evaluation of security properties of IT products.

3.1.2.9 IEC 60204-1 Safety of machinery wrt. electrical equipment of machines

IEC 60204-1:2016[10] applies to electrical, electronic and programmable electronic equipment and systems to machines not portable by hand while working, including a group of machines working together in a co-ordinated manner. The equipment covered by this part of IEC 60204 commences at the point of connection of the supply to the electrical equipment of the machine.

3.1.2.10 IEC 62433: Security for Industrial Automation Control Systems (IACS)Case study I. – Swarm of drones

IEC 62433-1:2019(E)[11] specifies the framework and methodology for EMC (Electromagnetic compatibility) IC (Integrated Circuit) macro-modelling. Terms that are commonly used in IEC 62433 (all parts), different modelling approaches, requirements and data-exchange format for each model category that is standardized in this series are defined in this document.

3.1.2.11 EU Reg. 2018/1139 on common rules in the field of civil aviation and establishing a European Union Aviation Safety Agency, and amending Regulations

The rules[12] are designed to ensure a common level of safety across the EU and give operators and manufacturers the predictability to develop products and services. Currently, drones lighter than 150kg fall under the jurisdiction of national authorities, with local manufacturers and operators being subject to different design and safety requirements. Under the new rules, Unmanned aircraft systems (UAS)s need to be

[7] https://www.iso.org/standard/53820.html
[8] https://www.iso.org/standard/59970.html
[9] https://www.iso.org/standard/50341.html
[10] https://webstore.iec.ch/publication/26037
[11] https://webstore.iec.ch/publication/59660
[12] https://www.easa.europa.eu/document-library/regulations/regulation-eu-20181139

designed and used in a way that they do not put people at risk. Drone operators must know the rules governing their flights and must demonstrate the ability to operate a drone safely, without putting people or other airspace users at risk. Some operators will therefore be required to go through training before they can operate a drone.
Operators of drones above 250g will need to be on national registers and their drones marked for identification. This is intended to allay privacy concerns and to assist in investigation in the event of an incident.

These amendments are intended to create an overall safety aviation regime, which is more fit for purpose, more proportionate and – crucially – risk-based. One of the key objectives is to handle better the expected increase in air traffic in the coming decades generally, while accommodating disruptive elements such as drones. UAS are likely to entail far greater numbers of individual flights: according to the EU Commission, civil drone technology could account for an estimated 10% of the EU aviation market within the next 10 years – about €15 billion per year – and create some 150,000 jobs in the EU by 2050. Drone activity will also higher levels of automation, which poses challenges as well as opportunities for the future world of unified traffic management. The risk-based approach also acknowledges that a lighter touch may be appropriate for leisure and sport aviation than for commercial air transport. Other changes on assessment of risk to flights over conflict zones and access to real time flight recorder data are clearly designed to minimise the recurrence of recent tragedies.

3.1.2.12 EN9100:2018

This International Standard[13] specifies requirements for a quality management system when an organization:

a) needs to demonstrate its ability to consistently provide products and services that meet customer and applicable statutory and regulatory requirements and

b) aims to enhance customer satisfaction through the effective application of the system, including processes for improvement of the system and the assurance of conformity to customer and applicable statutory and regulatory requirements.

3.1.2.13 ISO 26262 Road vehicles — Functional safety

ISO 26262[14] is an international standard for functional safety of electrical and electronic systems in production automobiles. Functional safety features form an integral part of each automotive product development phase, ranging from the specification, to design, implementation, integration, verification, validation, and production release. It defines functional safety for automotive equipment applicable throughout the lifecycle of all automotive electronic and electrical safety-related systems.

As a careful reader might have noticed, all of the above standards in robotics describe safety, but not cybersecurity. This is due to the fact that in the past cyber-security in the robotics industry was not a major issue, since robots and their controllers were not connected to the outside world in any meaningful way, let alone to the internet. However, the industry is changing and there is a push to connect these machines to the internet. The robotics industry has to adapt and maybe tailor ISO and American National Standards Institute (ANSI) standards on cybersecurity – which have been developed over decades and are already mature - in order to adjust industrial requirements for the era of CPSs.

[13] https://webstore.ansi.org/standards/din/dinen91002018
[14] https://www.iso.org/standard/68383.html

4 Case studies

4.1 Case study I. – Swarm of drones

4.1.1 Scenario overview

This section is a shorter version of the use-case description in D8.1 Initial Swarm of drones and ground robots demonstration .

In CPSwarm, the partners considered heterogeneous swarms of ground robots and UAVs (quadcopter) that, in a collaborative and completely autonomous way, scan a large outdoor area searching for human victims or people trapped in the disaster area. In a Search And Rescue (SAR) scenario, swarms can be exploited for:

a) generating a situation overview of the disaster scene in case of an industrial plant accident including real-time images (VIS, IR (Visible Infrared)), toxic and explosive gas leakage detection;
b) finding human casualties or people trapped in the disaster area.

The gathered information is used to help security personnel, first responders as well as rescue teams to conduct their mission efficiently. This application scenario has some fundamental requirements: a vast spatial area has to be inspected and information has to be provided to the stakeholders (security personnel, rescue teams, etc.) in real-time, especially in case of an incident. Swarms can reduce the inspection/detection times compared to, e.g., single UAV/rover applications due to their super-linear characteristics (the effect of the overall system is more than the sum of the effects of its individual parts). A concept image can be seen in Figure 8.

Figure 8 - Use case scenario (e.g. industrial plant)

4.1.2 Hardware specification for the demonstration

The following system architecture has been adopted both for quadcopters and for rovers, thanks to the autopilot's flexibility and the ROS-based companion computer, which have made it possible to keep the software almost completely unaltered and changing only parameters depending from the hardware.

Figure 9 and Figure 10 show the hardware in more detail.

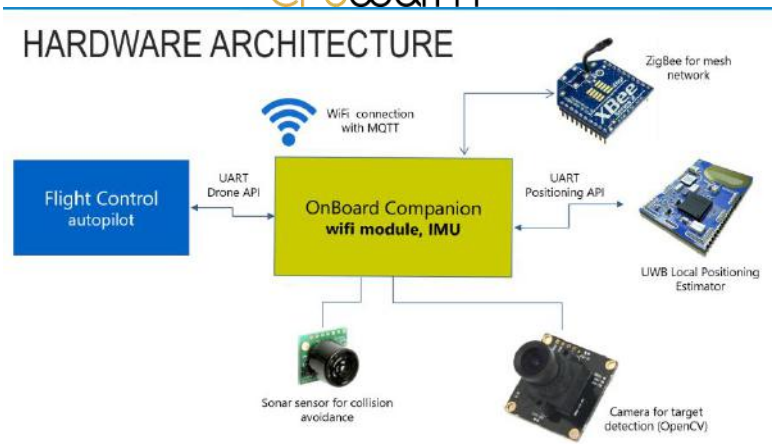

Figure 9 - Main components of quadcopter/rover

Figure 10 - Hardware features

4.2 Case study II. – Automotive

This section is a shorter version of the use-case description in D8.5 Initial Automotive Demonstration.

4.2.1 Vehicle Platooning Concept

- The leading vehicle has autonomous driving capability and prescribes the actions and decisions (i.e. navigation, decision on take-over maneuvers, sequencing maneuvers, lane change etc.) for the follow-up vehicles.
- The follow-up vehicles have autonomous driving capability and environmental awareness, too, to be able to react on specific driving scenarios requiring separate action (i.e. lane change and not enough space in new lane due to heavy traffic). In general, they follow the leading vehicle´s actions (see Figure 11).

Figure 11 - Vehicles in platooning configuration.

4.2.2 Architecture

Each vehicle in the platoon will be equipped with the same capabilities whether realized by the same components or other providing the same functionality as a prerequisite for such application (see Figure 12).

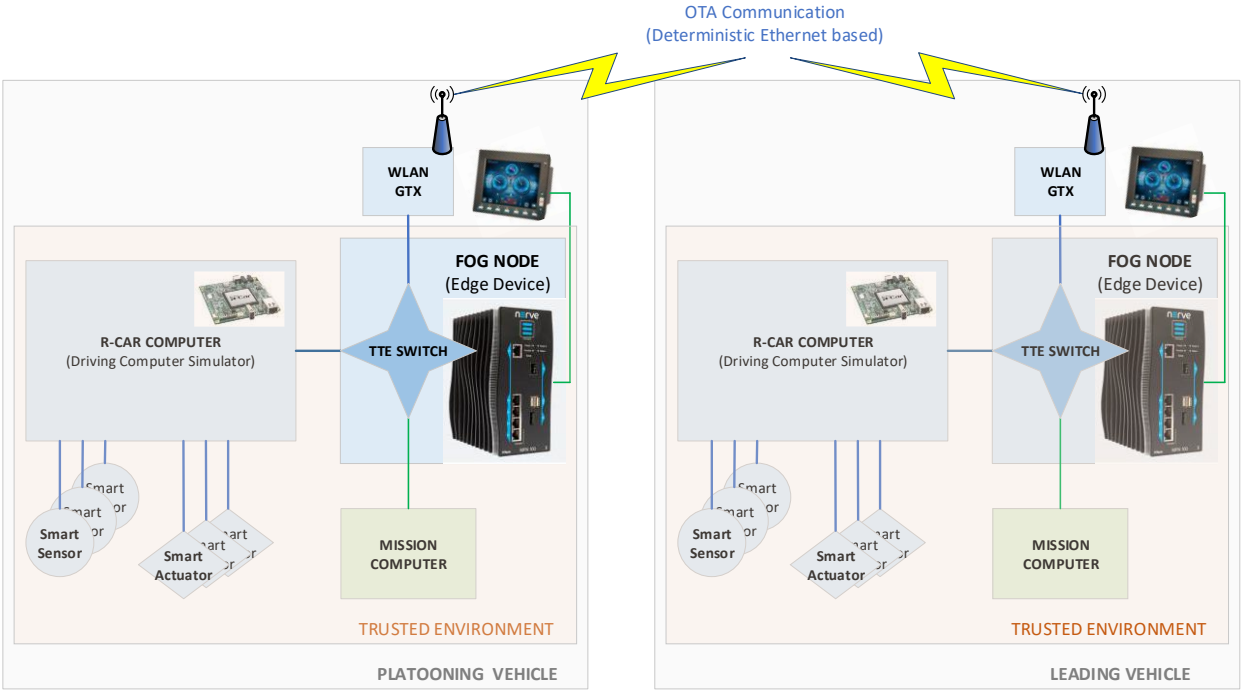

Figure 12 - Architectural set-up of the automotive use case

4.2.3 Deterministic wireless driver

Autonomous vehicles can only communicate with each other over the air (wireless) while they run on the road. The challenge therefore is to apply the know-how of the wired Deterministic/TTEthernet on a wireless environment (as illustrated in Figure 13). Deterministic/TTEthernet is a scalable technology and allows development of critical system parts according to fail-safe or fail-operational application requirements.

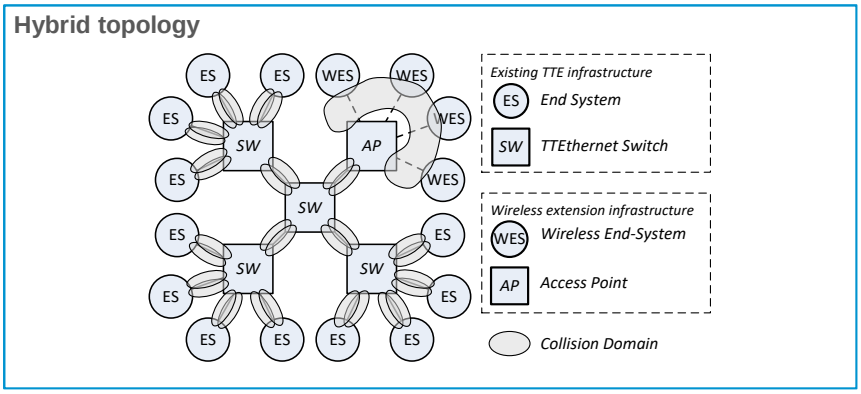

Figure 13 - TTEthernet topology

The main difference between the wired and wireless links is that wireless constitute a single collision domain when the stations are in range, whereas the wired links are full-duplex.

In order to support integration of applications with different real-time and safety requirements in a single network, Deterministic/TTEthernet supports three different traffic classes:

- time-triggered (TT) traffic - is sent in a time-triggered way, i.e., each Deterministic/TTEthernet sender node has a transmit schedule, and each TTE-Switch has a receive and forward schedule. This traffic is sent over the network with constant communication latency and small and bounded jitter.
- rate-constrained (RC) traffic - is sent with a bounded latency and jitter ensuring lossless communication. Each TTEthernet sender node gets a reserved bandwidth for transmitting messages with the RC traffic. No clock synchronization is required for RC message exchange (not used in the demonstrator, mainly implemented for aerospace applications).
- best-effort (BE) traffic - traffic with no timing guarantees. BE traffic class compatible with the IEEE 802.3 standard Ethernet traffic (will be used for the mission computer data communication).

4.2.4 Challenges of the automotive scenario

1. Wireless communication

   The communication from the leading vehicle to the follower vehicles, and also among all platoon participating vehicles as well as those intending to join the platoon, must mandatorily be wireless since it is not possible to have a wire among vehicles when they are running in a realistic situation.

2. Real-Time communication

   Real-time communication is compulsory for all safety/security related data communication (i.e., in the use case all autonomous driving related communication) to give response to the safety requirements, for example, when breaking. Network communication technology must use time scheduling to implement deterministic real-time communication.

3. Low reliability communication

   Real circumstances like harsh weather conditions, obstacles or presence of other wireless signals may decrease the reliability of wireless transmissions and can compromise real-time communication requirements. Considering that the quality of the wireless channels varies with the time, frequency and location, it is possible to increase reliability by finding better times, frequencies and locations to transmit and/or by performing retransmissions, while still obeying deadlines.

4.3 Case study III. – Logistics

This section is a shorter version of the use-case description in D8.3 Initial Swarm Logistics Demonstration .

The Swarm Logistics scenario involves robots, rovers and drones that collaboratively perform opportunistic scanning of the warehouse (see Figure 14). The idea is to scan the entire area of the warehouse and share the acquired information to update the knowledge base on the go. In addition to collecting information about the maps of the entire area, the connected robots will also be used for collecting additional information implicitly e.g. room temperature, presence of humans, detection of in-path obstacles etc. Since all the connected robots of the swarm acquire the information collaboratively, the status of the area is always up to date and the effort is always divided among all members. As a starting point, each connected robot will be fed with some default information, e.g., map of the area. This information is updated opportunistically on the go as the robots perform their main tasks. The main tasks of the robots are intended to assist humans in the logistics domain. These assistive tasks could include joining forces to move a heavy obstacle from one place to another.

Figure 14 - Impression on the Swarm Logistic scenario.

4.3.1 Hardware specification for the demonstration

To demonstrate the use case scenario, a robot model has been built from scratch. The base of the robot model is a iClebo Kobuki robot. It is a low-cost mobile research base designed for education and research on state of art of robotics.

Three different levels of height with three hexagonal plates have been designed to be on the top of the Kobuki base as Figure 15shows. On the first layer, an RPLidar A2 model has been integrated to have the ability to detect objects or obstacles. This laser has a range of 18 meters taking 4000 measures at 10 Hz frequency. On the second layer, an Intel NUC i5 with 8 Gb of RAM and 128Gb of SSD has been installed to work as the brain of the platform(Figure 16). This mini-computer offers Wi-Fi (802.11ac), Bluetooth 4.2 and Ethernet interfaces to interconnect the hardware elements and to communicate with the others CPS. On the third floor of the platform, the linear actuator CAHB-10 from SKF has been integrated to move the top plate of the robot to move up and down the last plate and to pick up and place things.

Figure 15 - Scout and carrier robots

Also, multicolor LED lights controlled by an Arduino board have been integrated to provide visual feedback about the state of the robot. Sensors on the robot are the following:

- Laser: Hokuyo UST-10LX
- camera: FLIR Chameleon CM3
- buttons: Schneider ZBRRC and ZBRT based on Zigbee Green Power
- Advantech ADAM 6060 I/O module supporting Modbus/TCP, TCP/IP, UDP, HTTP, DHCP, SNMP, MQTT

Figure 16 - Architecture of components connections

4.3.2 Challenges of the swarm logistics scenario

It is important to remark that the developments within this use case and demonstration respond to an actual need in the framework of mobile robotic logistics. The assignment of tasks according to different situations, the procedures and individual behaviour that each robot is driven with, the interaction and commands received from an operator and the way the system handles all of these are an interesting exercise for common problematics in our day to day work.

And then not only for the operational layer but also for the overall picture, this kind of autonomous work for the robots (to continuously empty an area) is really interesting yet not fully developed as a whole in Robotnik's system. We could highlight here that normally, even if the common tools and subsystems are reused, there is always a need of some coding or development per case according to the user needs.

5 Safety aspects

Security for Industrial Automation Control Systems (IACS) defined by international-generic (specifically ISA 99/IEC 62443[15]) and segment-specific standards (e.g., NERC =North American Electric Reliability Corporation) for United States power grid and management systems) regulate life-cycle management, compliance and recommended/required best practices.

Aspects to IACS security:

- Establishing a framework/process/program to create and manage IACS
- Aspects of IACS use cases and life-cycle (design and development, deployment, operation and maintenance)
- Conformance metrics for IACS security on system and component level
- Technical security requirements for host systems, network components, applications and embedded devices
- Product development requirements

The definition of a CPS implies interaction with the real world and its inhabitants – people. As a result, the safety of such systems is paramount. The following subsections provide an overview of the three scenarios envisioned in the CPSwarm project.

5.1 Safety considerations

Although the implementations of the CPSwarm use cases are in their infancy, their safety requirements overlap with a great deal existing policy and research. This section provides a brief examination of safety developments in these areas. For ease of discussion, the following subsections consider policy related to flying conventional drones (an element of the SAR use case), autonomous vehicles (used both in the SAR and automotive use cases) and the integration of robots in the workplace (the logistics use case).

5.1.1 Drones

The simplest instructions for flying a drone come from the European Aviation Safety Agency. In the poster *Flying a Drone*[16] the Agency outlines several dos and don'ts universal to all drone classes. Firstly, it recommends that routine maintenance is performed and that drones are only used within the performance limitations defined by the manufacturer. Due to liability, all operators should ensure that they are adequately insured, drones are kept in sight, a safe distance is maintained to people, animals and property and other aircraft, and, in the event of an accident, the relevant national agency is informed. Drones should not fly higher that 120m above the ground. Conventional no-fly zones must be observed: drones should not be used near manned aircraft, airports, helipads, other restricted area, areas affecting public safety or where an emergency response is ongoing. Lastly, all relevant privacy laws must be respected; photos, videos or sound recordings should not be made without permission.

In addition to the guidelines above, the Agency has worked to define the concept of U-Space[17]: a set of services designed to facilitate the operation of drones below 120m. The key components of drone registration, electronic identification and geo-awareness should provide air traffic management, safe operation and ultimately ensure that drones do not enter any restricted areas[18]. In theory, when the system is operational in 2025, it should allow fully-autonomous drones to fly beyond the line of sight of the operator.

[15] https://webstore.iec.ch/publication/7029
[16] https://www.easa.europa.eu/sites/default/files/dfu/217307_EASA_DRONE_POSTER_2018%20final.pdf
[17] https://www.sesarju.eu/sites/default/files/documents/reports/U-space%20Blueprint%20brochure%20final.PDF
[18] https://www.easa.europa.eu/sites/default/files/dfu/217603_EASA_DRONES_LEAFLET%20%28002%29_final.pdf

5.1.2 Automated vehicles

The safety of highly automated vehicles came into the spotlight following two fatalities in 2018: the first involving a Level-4 (high driving automation) autonomous taxi and the second involving a Level-2 (partial driving automation) autopilot prompted calls for autonomous vehicles to pass minimum standards for safety, reliability and performance. Efforts such as the University of Michigan's Mcity ABC Test[19] aim to win back trust for such systems. The test consists of three parts: accelerated evaluation, behavior competence and corner cases.

Accelerated evaluation aims to consider a system's performance in the most common scenarios involved in a crash, namely following a car, changing lanes and making left turns. Particular importance is given to a system's reaction to risky behavior.

In behavior competence systems must demonstrate their ability to handle common scenarios. The Mcity ABC Test includes 50 scenarios based on the findings of 8 leading research groups, 35 of which were identified as responsible for major crashes. Importantly, autonomous vehicles must also be tested in a spectrum of different weather conditions and lighting levels. While this is cost prohibitive, the researchers suggest testing vehicles during day and night time, as well as testing the perception systems in rain and snow.

Finally, corner cases include tests of extreme conditions that vehicles may experience in the real world. They are of particular value in exposing system weaknesses. One such example given by the researchers is a camera system's ability to recognize a black car at night.

Within Europe, legislation always requires the presence and attention of drivers in autonomous vehicles (Vienna Convention, ECE R79). Furthermore, automatically commanded steering functions are only permitted for vehicles travelling up 10 km/h. Thus, systems also must consider how they maintain driver awareness and how control is handed over between the autonomous system and the driver[20].

Additionally, as autonomous systems operate in a space shared with other potentially irrational agents, they will be involved in incidents. In such situations safety tradeoffs must be made. For example, the system may have to prioritize the safety of its passengers over other parties. In the *Moral Machine*[21] survey by Awad et al., significant differences were recorded between regions around the world. For example, people from Asian countries with strong Confucian or Islamic traditions were less likely to sacrifice older people to protect the young than the other two regions in the study. Arguably the reasoning behind decisions and consideration of any associated bias may also be important in fostering trust in automated systems.

5.1.3 Industrial Robots

Many accidents involving robots occur outside their normal operating conditions, for example, during programming, maintenance, testing or setup. In such situations workers may be interacting with robots inside areas they would normally be excluded from or more likely to produce unintended operations[22].

To minimize the risk of accidents, the National Institute for Occupational Safety and Health, part of the U.S. Department of Health & Human Services, recommends that safety be considered during the design of robotic systems, worker training and worker supervision[23].

During design, physical barriers should ensure that workers are unable to come into direct contact with moving parts of the robot. Sensors should be integrated into the system to automatically stop. Systems

[19] https://mcity.umich.edu/wp-content/uploads/2019/01/mcity-whitepaper-ABC-test.pdf
[20] https://wiki.unece.org/download/attachments/25267488/ACSF-01-11%20-%20%28J%29%20concept%20paper.pdf?api=v2
[21] http://moralmachine.mit.edu/
https://www.nature.com/articles/s41586-018-0637-6
[22] https://www.osha.gov/SLTC/robotics/index.html
[23] https://www.cdc.gov/niosh/docs/85-103/

should be designed to support the maximum degree of remote diagnostics, minimizing direct worker-robot contact. Lastly, the working areas of the robot as well as the location of safety shutoff systems should be clearly marked.

Importantly, all workers should receive training about the robots, its programming, range of motion, safety systems and correct operating and maintenance procedures. The speeds and areas of operation should be fitted to the actual operating conditions. Finally, continued supervision of workers is required to ensure that safety procedures function as intended and that they are followed over the long term.

6 Assets

6.1 Assumptions

This chapter first describes assets which are common for all of the use cases of the CPSwarm project. Use case specific assets will be described and identified in the next sections. First some common assumptions are presented, which hold for all of the CPSwarm use-case scenarios and moreover, can cover other use cases of swarms of CPSs.

- A swarm of robots is deployed in a mission, which requires information gathering and/or processing;
- Swarm members are equipped with different sensors, cameras and GPS modules;
- Swarm members can communicate with each other and/or the operator of the mission;
- The operator of the swarm initializes the mission by defining the objectives, targets and area of operation, and monitors the swarm remotely;
- External authorities such as police or border control may communicate remotely with the swarm in case of emergency or violation of local policies.

6.2 Generic assets

Having these assumptions in mind, we can now define the generic assets. Assets can be grouped into two categories: tangible and intangible assets. The intangible asset categories that should be protected when deploying a swarm of CPSs are

- *Information* - gathered and/or possessed by the swarm, including intellectual property and mission parameters,
- *Service* - meaning the capability to successfully execute the mission, includes the operability and performance characteristics of individual swarm members, and
- *Environment* - including safety and non-disruption to objects, humans and animals concerning the swarm's operation site.

The CPSwarm Consortium has identified the primary intangible assets in Table 1.

ID	Asset	Category
PA1	Operational parameters *Commands and additional information supplied by the operator to govern the behaviour of the swarm, including the area of deployment, the location of targets, etc.*	Information
PA2	Data gathered by swarm members *Any information collected by on-board sensors, including audio-visual feeds, component status and location, as well as information received from other swarm members or generated locally.*	Information
PA3	Swarm algorithm *The generic algorithm used by the swarm to solve the problem, which might be used to predict or sabotage swarm behaviour or might have*	Information

	significant market value.	
PA4	Presence of the swarm *The very fact that the swarm is operating in the vicinity.*	Information
PA5	Goal *The ability of the swarm to solve the specified problem and only the specified problem.*	Service
PA6	Controllability *The operator's ability to issue new commands, specify goals and in general maintain control over the swarm and its members.*	Service
PA7	Performance *The ability of the swarm to maintain the expected timeliness, manner and quality of service required to solve the task as expected by the operator, including the continuous feasibility of redeployment.*	Service
PA8	Environmental non-disturbance *The ability of the swarm to operate in a manner that does not disturb or interfere with the natural and manmade environment – operating such that no property damage occurs, the disturbance of bystanders is minimized and the natural environment is respected.*	Environment
PA9	Safety *The ability of the swarm to guarantee the safety of humans inside and outside its operational area.*	Environment
PA10	Compliance *The continuous assurance that the swarm and its members operate within the confines of the law and any applicable standards.*	Environment

Table 1 - Primary assets

The secondary, tangible assets are what can be protected - they support the primary, intangible assets and might possess vulnerabilities, which can be exploited by an attacker aiming to corrupt the intangible assets - are the agents including their hardware and software components and operators including the personnel and the system they are using when setting up and monitoring the swarm. The tangible asset categories we would like to protect are

- *Software* – the operating system and software controlling the robots,
- *Hardware* – the body of the robot, including sensors and their perceived reality.
- *Personnel* – the operators, developers and other privileged members of the organization responsible for the development and deployment of the swarm.

- *Data* – data required for carrying out a successful mission (sensor data gathered, mission objective, description, map, cryptographic keys)

The protection of personnel assets is outside the scope of this document (and the project). With that said, the following secondary assets have been identified in the other categories in Table 2.

ID	Asset	Category	Concrete asset
SA1	Swarm member firmware *The operating system and embedded base software of individual swarm members that serve as the platform for the rest of the software.*	Software	Embedded Linux variants (FogOS, ROS)
SA2	Swarm algorithm implementation *The concrete implementation of the swarm algorithm as designed and deployed by the operator.*	Software	Use case specific
SA3	Operator software environment *The operating system and software that is present on the systems used by the operator.*	Software	Linux
SA4	Operator toolset *A set of software tools used by the operator to interact with the swarm and its members.*	Software	CPSwarm Deployment Tool CPSwarm Monitoring Tool
SA5	Communication protocol implementation *The implementation of the communication protocol used by either the operator toolset or the swarm algorithm implementation.*	Software	CPSwarm Communication Library
SA6	Locomotion *The ability of the CPS to move around at will in physical space.*	Hardware	Aerial or ground robots
SA7	Sensing *The hardware components responsible for gathering information about the environment and the CPS itself, including the correctness of such observations.*	Hardware, Data	Use case specific
SA8	Actuation *The hardware components responsible for interacting with the environment and the control data.*	Hardware, Data	Use case specific

SA9	Consumables *Any resource that is being consumed while the swarm is in operation – including battery charge, fuel, coolant, etc.*	Hardware	Use case specific
SA10	Processing *Computing resources – CPU time, RAM, etc. – that are required to execute the swarm algorithm and any dependent processes.*	Hardware	Use case specific
SA11	Structural integrity *The ability of the CPS to maintain its physical integrity throughout the mission.*	Hardware	Use case specific
SA12	Connectivity *Refers to the components that make up the physical layer of communications, including any radios, cabling, visual signage, etc.*	Hardware	Use case specific
SA13	Operator hardware environment *The underlying hardware that is used by the operator to run tools and control the swarm.*	Hardware	PC or notebook with network access (VPN/APN)
SA14	Cryptographic secrets *Private keys used for communication: authentication, authorization and integrity protection, firmware signing, encryption*	Data	Communication Keys
SA15	Mission data *Sensitive data sent to mission control and other swarm members*	Data	GPS trail, telemetry

Table 2 - Secondary assets

6.3 Analysis of relevant assets

One of the key concepts in information security is the CIA triad – Confidentiality, Integrity and Availability. They can be described in terms of what they could mean in the context of a mission that uses swarms of CPSs. Specifically:

1. *Confidentiality* means the non-disclosure of sensitive data to unauthorized parties – for instance the data collected or initially stored by the CPS, software and swarm algorithms and mission parameters.
2. *Integrity* means that sensitive data cannot be modified without authorization – e.g., data possessed by swarm members, software used in the runtime environment, mission parameters or emergency signals.
3. *Availability* means that the system, its components or certain functionalities must be available to operate when needed.

In this section, the authors study whether the confidentiality, integrity and availability of each asset can be compromised. In Table 3 - CIA analysis of assets, the results are summarized according to the following legend:

- Y – yes
- N – no
- N/A – not applicable

ID	Asset	C	I	A
PA1	Operational parameters	Y	Y	Y
PA2	Data gathered by swarm members	Y	Y	Y
PA3	Swarm algorithm	N	Y	Y
PA4	Presence of the swarm	Y	N/A	N/A
PA5	Goal	Y	Y	Y
PA6	Controllability	N/A	N/A	Y
PA7	Performance	N/A	N/A	Y
PA8	Environmental non-disturbance	N/A	N/A	Y
PA9	Safety	N/A	N/A	Y
PA10	Compliance	N/A	N/A	Y
SA1	Swarm member firmware	N	Y	Y
SA2	Swarm algorithm implementation	N	Y	Y
SA3	Operator software environment	N	Y	Y
SA4	Operator toolset	N	Y	Y
SA5	Communication protocol implementation	N	Y	Y
SA6	Locomotion	N/A	N/A	Y
SA7	Sensing	Y	Y	Y
SA8	Actuation	N/A	Y	Y
SA9	Consumables	N/A	N/A	Y
SA10	Processing	N/A	N	Y
SA11	Structural integrity	N/A	N/A	Y

SA12	Connectivity	N/A	Y	Y
SA13	Operator hardware environment	N/A	Y	Y
SA14	Cryptographic secrets	Y	Y	Y
SA15	Mission data	Y	Y	Y

Table 3 - CIA analysis of assets

Confidentiality, where required, is always treated as use-case specific. In certain use cases, confidentiality can even be something that is explicitly forbidden or detrimental to the operation of the swarm member or the swarm. Nonetheless, where confidentiality is a potential requirement, the possible attack paths will be explored in this document.

The following sections map the general assets presented in the previous section to specific assets of the use-cases.

6.4 Use case I. specific assets

The assets specific to the swarm of drones scenario is summarized in Table 4.

ID	Asset	Category	Concrete asset
SA1	Swarm member firmware	Software	Embedded Linux variant (ROS), PX4 Flight Stack, PX4 Middleware, OpenCV, QGroundControl
SA2	Swarm algorithm implementation	Software	No manual control, Safe landing, Separated flight control and business logic
SA3	Operator software environment	Software	Linux
SA4	Operator toolset	Software	CPSwarm Deployment Tool CPSwarm Monitoring Tool
SA5	Communication protocol implementation	Software	CPSwarm Communication Library, Message Queue Telemetry Transport (MQTT)
SA6	Locomotion	Hardware	Aerial (rotors) and ground robots (wheels) with electric motors
SA7	Sensing	Hardware, Data	GPS, Camera, UWB, Ultrasound Coordinates, elevation, distance, photos, videos captured
SA9	Consumables	Hardware	Battery
SA10	Processing	Hardware	SoC of the device (e.g. quad-core ARM Cortex-A7 1.2 GHz)
SA11	Structural integrity	Hardware	Bumpers, ultrasound sensors for collision avoidance

SA12	Connectivity	Hardware	WiFi, Cellular, Bluetooth, UWB, ZigBee
SA13	Operator hardware environment	Hardware	PC or notebook
SA14	Cryptographic secrets	Data	Communication Keys: WiFi, ZigBee, Communication Library (shared secret key, certificates, session keys)
SA15	Mission data	Data	Mission plan, target locations, safe passage, swarm member location, telemetry, battery level, camera feed

Table 4 - Assets for use case I.

6.5 Use case II. specific assets

The assets specific to the automotive scenario is summarized in Table 5.

ID	Asset	Category	Concrete asset
SA1	Swarm member firmware	Software	Embedded Real-time Linux variant, (Fog OS)
SA2	Swarm algorithm implementation	Software	Platooning behaviour, Emergency routine (mission abort)
SA3	Operator software environment	Software	Linux
SA4	Operator toolset	Software	CPSwarm Deployment Tool CPSwarm Monitoring Tool
SA5	Communication protocol implementation	Software	CPSwarm Communication Library TTEthernet
SA6	Locomotion	Hardware	Cars/Trucks
SA7	Sensing	Hardware, Data	GPS, Camera, Ultrasound, Smart sensors Coordinates, distance, photos, videos captured
SA8	Actuation	Hardware, Data	Smart actuators and control data
SA9	Consumables	Hardware	Battery, Fuel, Coolant, Oil
SA10	Processing	Hardware	FogNode, R-Car
SA11	Structural integrity	Hardware	Provided by vehicle chassis, ultrasound sensors
SA12	Connectivity	Hardware	WiFi, 3G/4G

SA13	Operator hardware environment	Hardware	PC or notebook
SA14	Cryptographic secrets	Data	Communication Keys: WiFi, Communication Library (shared secret key, certificates, session keys)
SA15	Mission data	Data	Mission route, current location, telemetry, battery/fuel level, camera feed

Table 5 - Assets for use case II

6.6 Use case III. specific assets

The assets specific to the logistics scenario is summarized in Table 6.

ID	Asset	Category	Concrete asset
SA1	Swarm member firmware	Software	Embedded Linux variants (ROS)
SA2	Swarm algorithm implementation	Software	Scouting and Carrier behaviours, Emergency routine (mission abort)
SA3	Operator software environment	Software	Linux
SA4	Operator toolset	Software	CPSwarm Deployment Tool CPSwarm Monitoring Tool
SA5	Communication protocol implementation	Software	CPSwarm Communication Library Modbus/TCP, TCP/IP, UDP, HTTP, DHCP, SNMP, MQTT, ZigBee, Bluetooth
SA6	Locomotion	Hardware	Ground robots with wheels, electric motor
SA7	Sensing	Hardware	Camera, Lidar, Ultrasound
SA8	Actuation	Hardware, Data	Elevator and control data
SA9	Consumables	Hardware	Battery
SA10	Processing	Hardware	Intel NUC i5, 8GB RAM
SA11	Structural integrity	Hardware	Bumper
SA12	Connectivity	Hardware	WiFi, Bluetooth, ZigBee
SA14	Cryptographic secrets	Data	Communication Keys: WiFi, ZigBee, MQTT, Bluetooth
SA15	Mission data	Data	Mission plan, member location, telemetry, battery level, camera feed, carried payload ID

Table 6 - Assets for use case III.

7 Attack trees

7.1 The anatomy of an attack tree

In this section, the authors briefly explain the visualization used for the attack trees using Modelio AttackTree plugin[24] as part of the CPSwarm Workbench.

As already mentioned in Section 2.1.6, an attack tree is a hierarchical diagram consisting of one root node, internal nodes, and leaf nodes. From the bottom up, nodes are conditions, which must be satisfied in order to make the direct parent node true. There are two types of parent nodes: *Type AND* and *Type OR*. In the first case, every child node must be satisfied; in the second case, at least one is required.

To enhance readability, the authors present smaller, narrower attack trees, thus the deliverable will detail specific attack types in separate trees, mostly in Section 7.4. In higher level attack trees, as in Section 7.2, it is denoted nodes, which are expanded in a separate tree as seen in Figure 17.

Figure 17 - Example for node expansion

The attacks are labelled as safety-related with blue font colour. The attacks marked with a red rectangle denote root attacker goals. Lower level attacks are marked with a yellow rectangle inside.

7.2 Attacker motivations

This section presents the high-level attack trees that describe the possible goals of the different malicious actors identified. Each of these will include an explanation of the motive and related attacks including the description of the affected primary assets.

[24] https://forge.modelio.org/projects/attack-tree-development/repository

7.3 General attacks and attacker goals

7.3.1 Sabotage mission

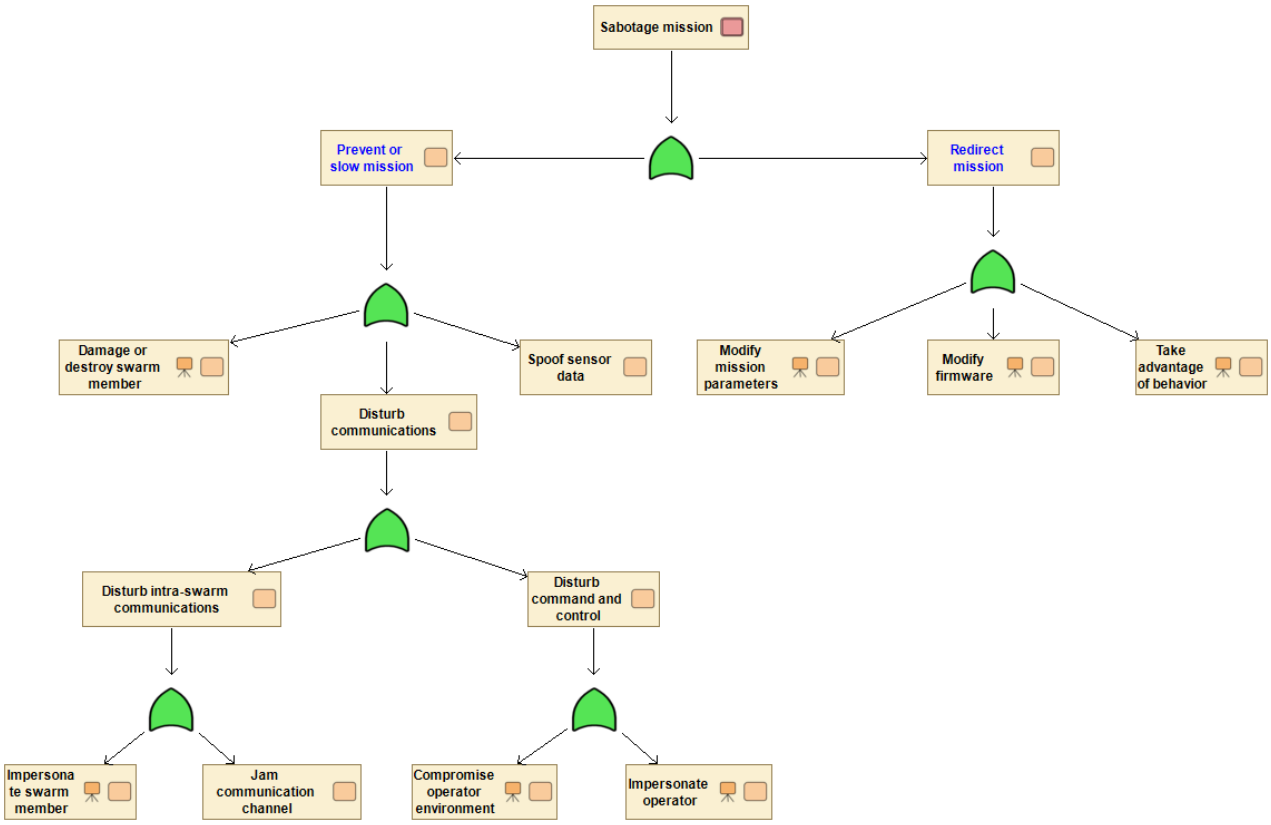

Figure 18 - Sabotage mission

The reasons behind mission sabotage (see Figure 18) can span from a simple prank to a serious act of warfare resulting in the loss of human lives - depending on the application of the swarm in question.

A sabotaged mission (either total or partial failure of it) can affect the majority of primary assets: PA1, PA2, PA3, PA5, PA6, PA7; while PA9 - Safety is affected based on the use case of the swarm, which will be assessed later. All the secondary assets can be affected depending on the application of the swarm and attack type.

7.3.2 Cause financial damage to swarm operator

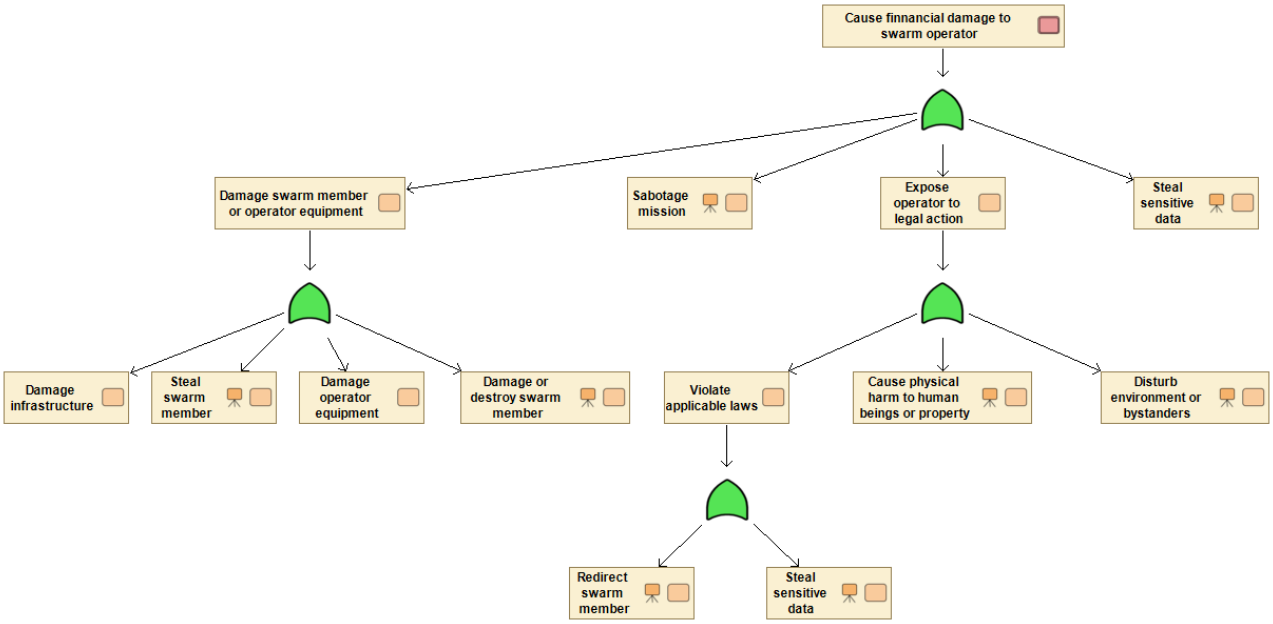

Figure 19 - Cause financial damage to operator

Financial damage to the swarm operator could be pursued by commercial competitors (to the operators) or criminals (see Figure 19). Not only physical damage or exhaustion of resources can lead to financial damage, but also violation of local regulations by the swarm or the theft of intellectual property owned by the operators.

Here the primary assets affected are the ones not connected to physical damage – PA8, PA9 and PA10, while the damage of swarm members or equipment concerns several secondary assets – SA3, SA4, SA5, SA6, SA7, SA9, SA11, SA13 and SA15.

7.3.3 Cause physical harm to human beings or property

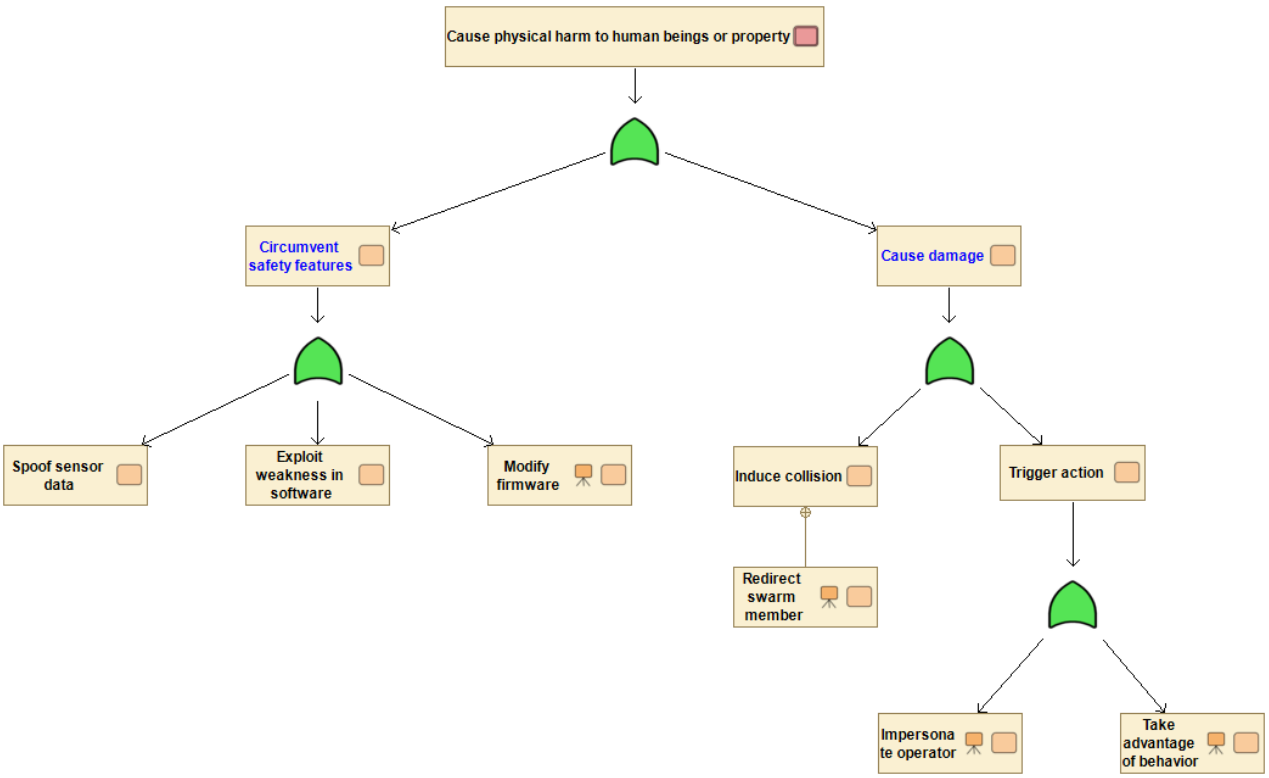

Figure 20 - Cause physical harm to human beings or property

Vandals and criminal organizations may desire to cause physical harm to private or public property, while terrorist groups could corrupt swarms in order to take as many lives as possible and to cause mass destruction. To successfully perform this, they not only need to find a way to possibly cause damage but also have to figure out how to circumvent safety features applied to the swarm (see Figure 20).

The affected primary assets are PA1, PA2, PA3, PA8 and PA9 while the secondary assets are SA1, SA2, SA3, SA4, SA7, SA8 and SA13.

7.3.4 Disturb environment or bystanders

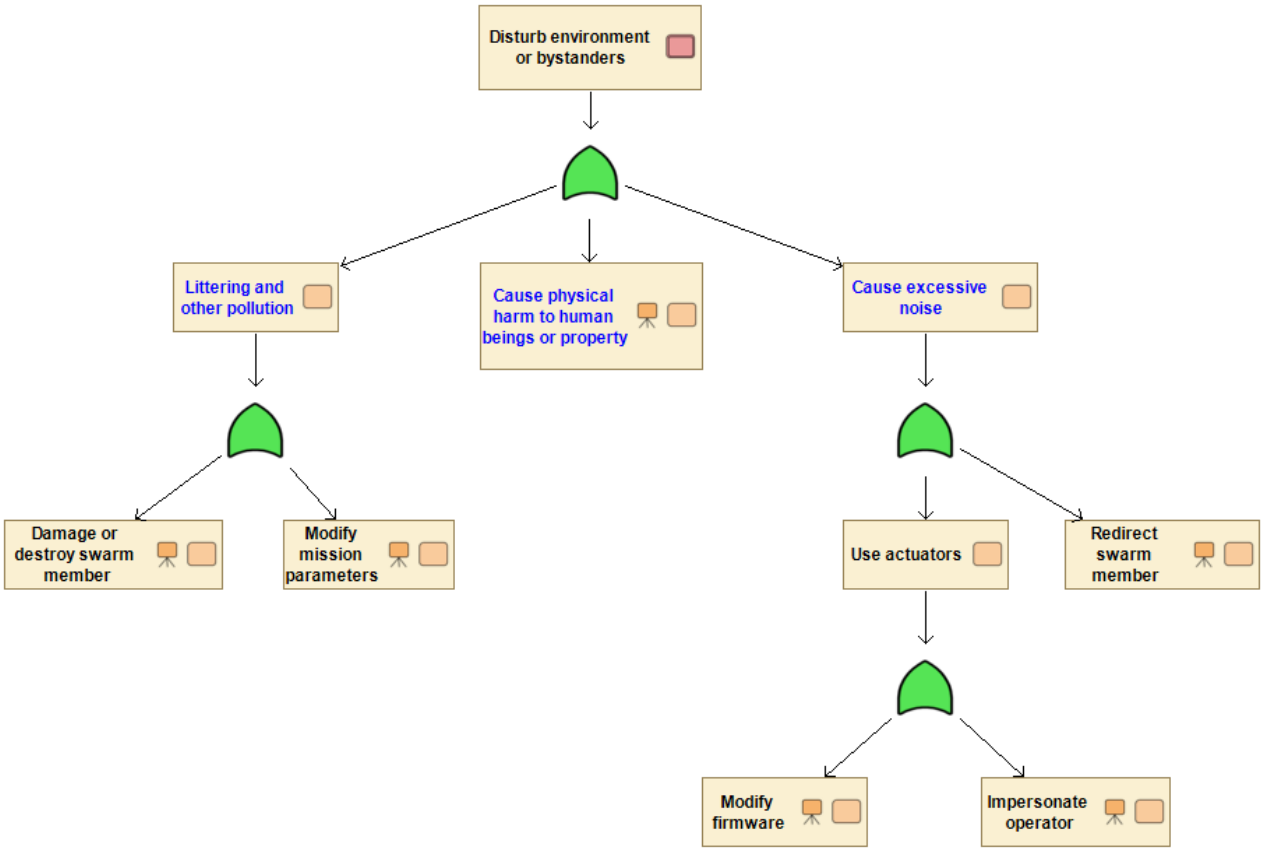

Figure 21 - Disturb environment or bystanders

Vandals may want to disturb the environment or humans just for fun, while more serious criminals would want people or local authorities to shift their attention to the disturbance caused by the swarm while they can commit other crimes simultaneously, for example theft or robbery (see Figure 21).

Means of destruction can also be used for distraction; hence the same primary and secondary assets are affected as in Section 7.3.3 (PA1, PA2, PA3, PA8, PA9, SA1, SA2, SA3, SA4, SA7, SA8 and SA13).

7.3.5 Steal swarm member

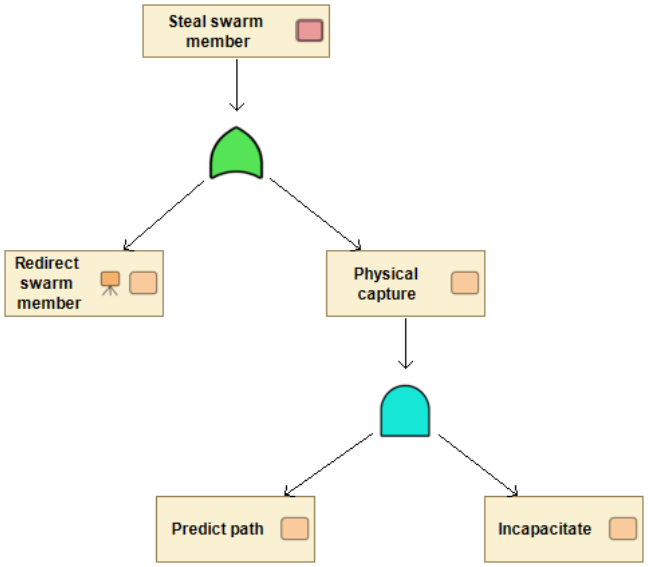

Figure 22 - Steal swarm member

Theft could be done for profit by criminals or maybe by vandals as a form of self-entertainment. Since it requires physical interaction with the swarm members, a thief can either redirect swarm members to a preferred location or physically capture them (see Figure 22) – lots of different ways of doing that have seen in the media lately, such as throwing nets or shooting blunt objects.

All primary and secondary assets are affected which can be associated with the physical entity of swarm members: PA1, PA2, PA3, PA6, SA1, SA2, SA3, SA4, SA5, SA6, SA7, SA8, SA9, SA10, SA11, SA12 and SA13.

7.3.6 Steal sensitive data

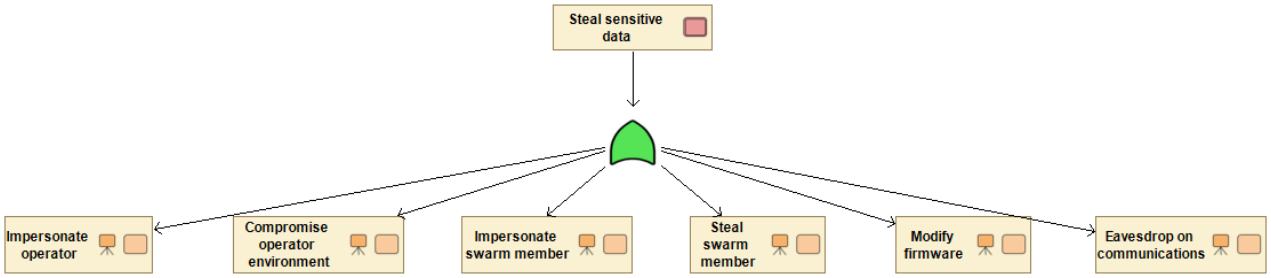

Figure 23 - Steal sensitive data

Sensitive data could be any data collected or possessed by the swarm which has to be kept secret. Similarly to stealing the physical devices, theft of sensitive data could be done for profit by criminals or by vandals for fun. In extreme cases, commercial competitors to the operators may want steal secrets connected to the swarm algorithms used or other intellectual property (see Figure 23).

Since this is a high-level attack tree, all the assets that are affected by the attacks described by the leaf nodes are affected here as well – all primary and secondary assets may be affected based on what is regarded a secret in the mission of the swarm in question.

The affected primary assets are PA1, PA2, PA3, PA4, and PA5, while the secondary assets are SA1, SA2, and SA14.

7.4 Methods of compromise

This section presents the lower-level attack trees that describe the possible actions an attacker has to accomplish to realize an attack. As seen in the previous section, a combination or selection of these could lead the attackers to realize their goal described above.

7.4.1 Damage or destroy swarm member

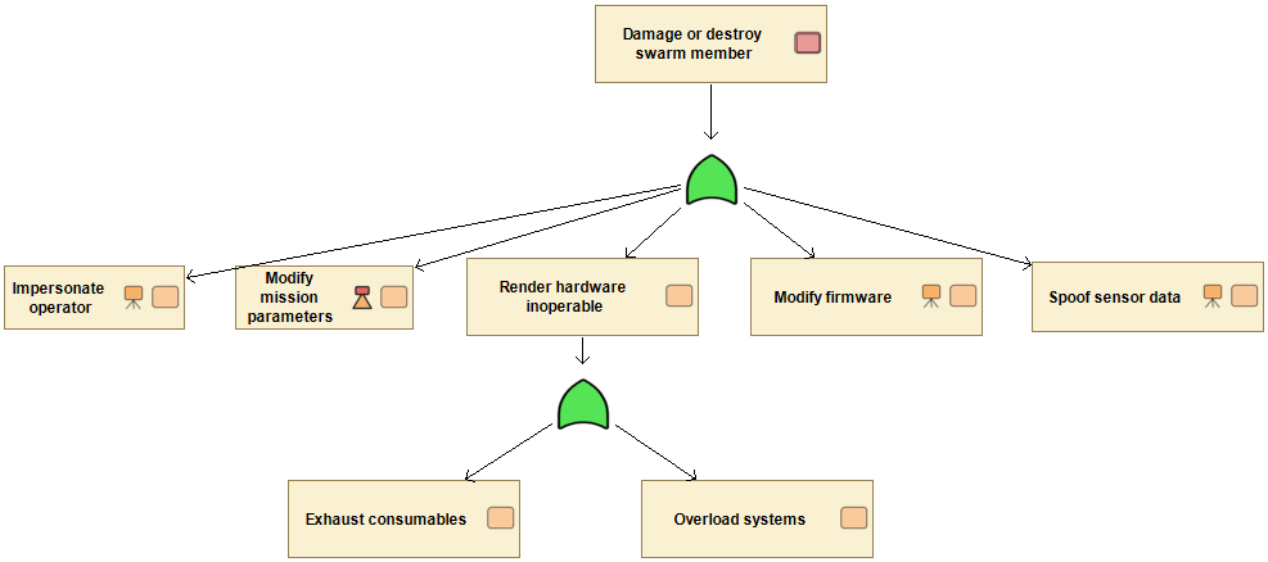

Figure 24 - Damage or destroy swarm member

Damaging a swarm member is both a way to cause property damage and to diminish the efficacy of the swarm. Apart from attacks that try to damage the property of others (and also involve damage to swarm member itself), the swarm member might be rendered inoperable without it ever being visible on the outside – either by bricking the software environment or by causing excessive wear on parts (see Figure 24).

Secondary hardware assets are the main targets of this attack (SA6, SA7, SA8, SA9, SA10 and SA11). Depending on the scope of the attack, the Goal and the Performance of the swarm might be diminished (PA5 and PA7).

7.4.2 Redirect swarm member

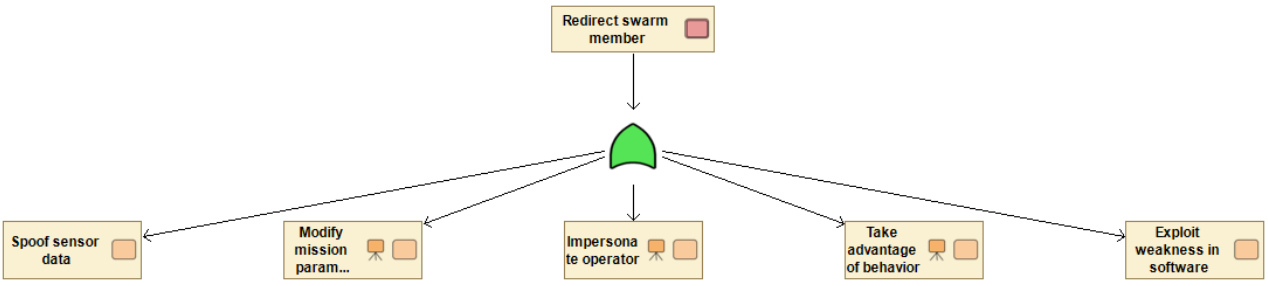

Figure 25 - Redirect swarm member

For many attacks, the attacker needs to move the swarm member to a desired location – which can be achieved by traditional attacks that target the command and control infrastructure, or through more creative means, which try to take advantage of the behaviour of the swarm member by spoofing targets, current location and other sensor data to get the swarm member to move to the right place at the right time on its own (see Figure 25).

This affects the primary assets in the Service category (PA5, PA6 and PA7), as well as this in the Environment category (PA8, PA9 and PA10). Of the secondary assets, Locomotion is targeted (SA6), but an attack may also affect others.

7.4.3 Take advantage of behaviour

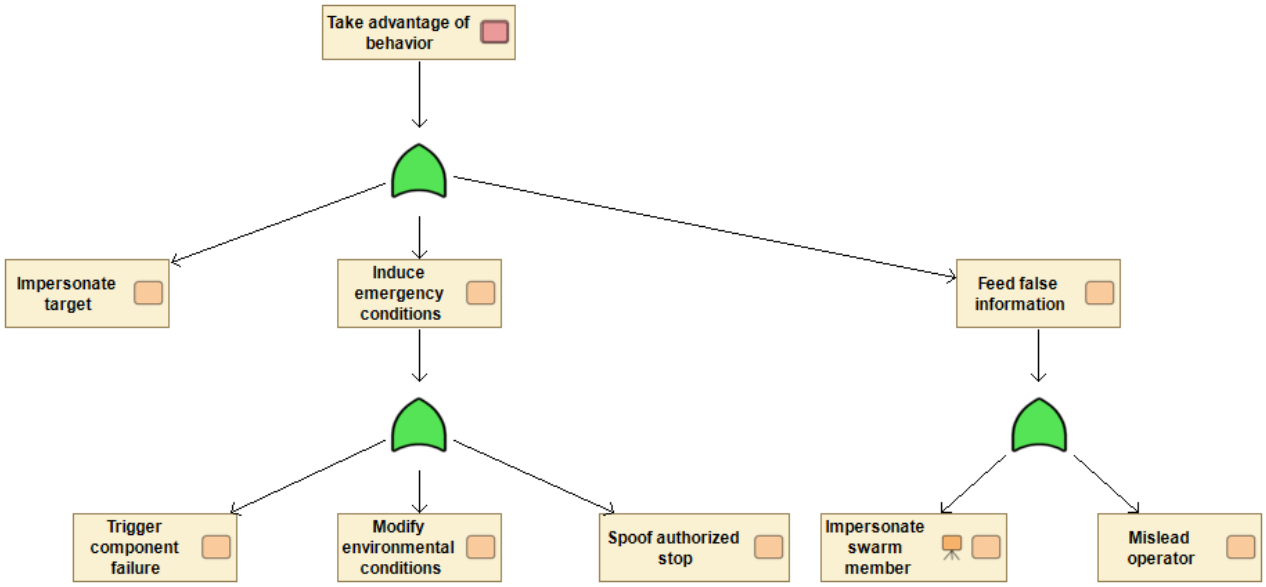

Figure 26 - Take advantage of behavious

If the behaviour of the swarm member follows a known pattern, an attacker can use that to its advantage by manipulating the environment in order to get the swarm to behave differently. This might include spoofing targets by reverse engineering the method used to identify them or feeding entirely false information either through the communication protocol or by providing misleading information to the operator. In certain cases, the attacker might also be interested in triggering an emergency condition – which may lead to the swarm member deactivating (see Figure 26).

This attack – depending on how it is used – can affect a great variety of assets. The Swarm Algorithm (PA3) needs to be already compromised if this attack is to be successfully mounted.

7.4.4 Modify mission parameters

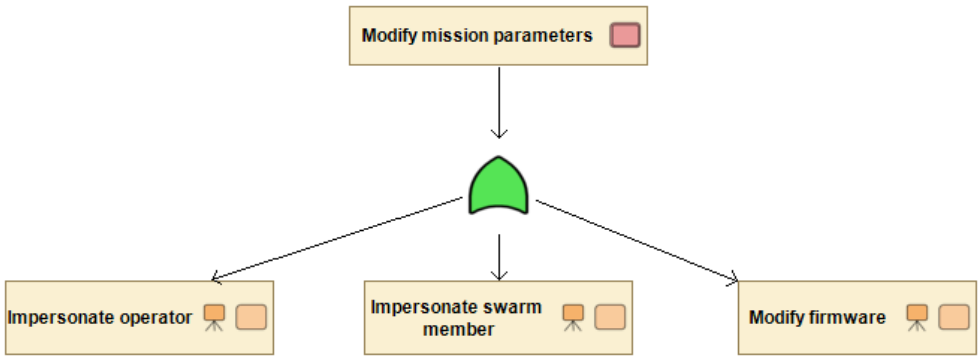

Figure 27 - Modify mission parameters

Mission parameters are set by the operator to define the boundary conditions under which the swarm operates. These might include the location of targets, the operational area or the distance to keep from dangerous objects. Since mission parameters might be updated on-the-fly, or might be changed as a result of the information provided by other swarm members, any compromise of the communication infrastructure can potentially lead to these parameters being changed (see Figure 27).

Modifying mission parameters can seriously affect the Goal and the Controllability of the swarm (PA5 and PA6), but might have consequences for other Service and Environment assets.

7.4.5 Eavesdrop on communications

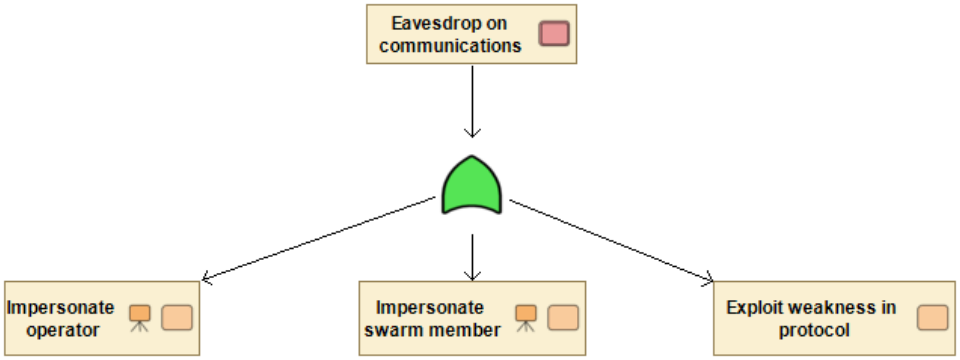

Figure 28 - Eavesdrop on communications

Eavesdropping requires a physical compromise of the underlying communication medium. Since in most cases, this refers to receiving radio signals, for which (depending on the exact protocol and technology) commercial equipment is widely available, the tree does not deal with gaining access to the medium itself. Performing this attack relies on the attacker's ability to either find a weakness in the protocol or to successfully act as a member or operator of the swarm. The term protocol refers to both the high level protocol used by the swarm and any other protocols the communication stack uses (see Figure 28).

A successful attack affects the confidentiality of all primary assets in the Information category (PA1, PA2, PA3, and PA4) and the Goal itself (PA5) – and of relevant secondary assets that might be transmitted over the link (SA1 and SA2).

7.4.6 Impersonate swarm member

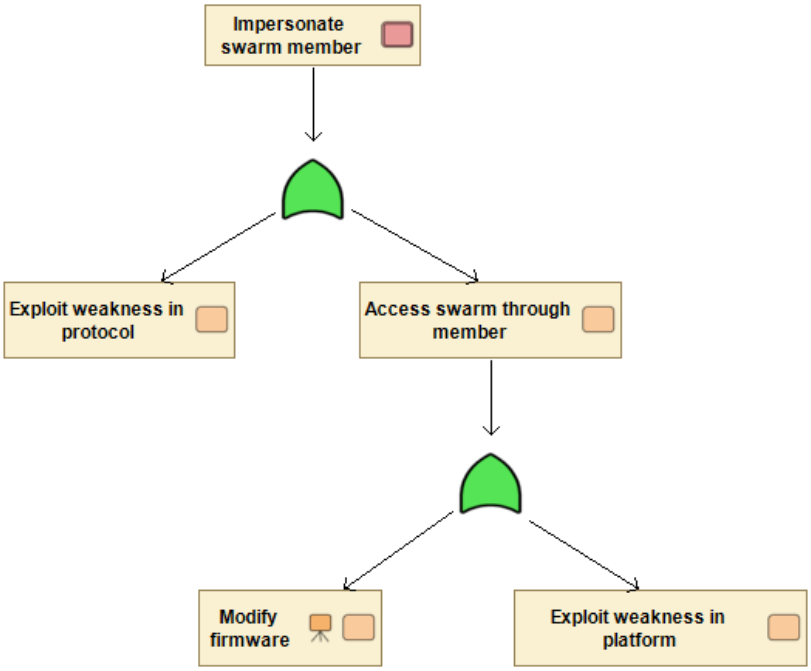

Figure 29 - Impersonate swarm member

To successfully impersonate a swarm member, the attacker has to be able to send and receive messages in a way that it is indistinguishable from another existing (or a newly introduced, non-existent yet nonetheless recognized and accepted) member of the swarm. If a weakness in the protocol is found, the attack can even

be performed without interacting with the swarm member in question – but a more likely scenario is that a swarm member is compromised in order to send and receive messages in its name (see Figure 29).

Such an attack affects the confidentiality of any information shared between swarm members can also compromise the integrity of information as received by the operator or other members (PA2).

7.4.7 Impersonate operator

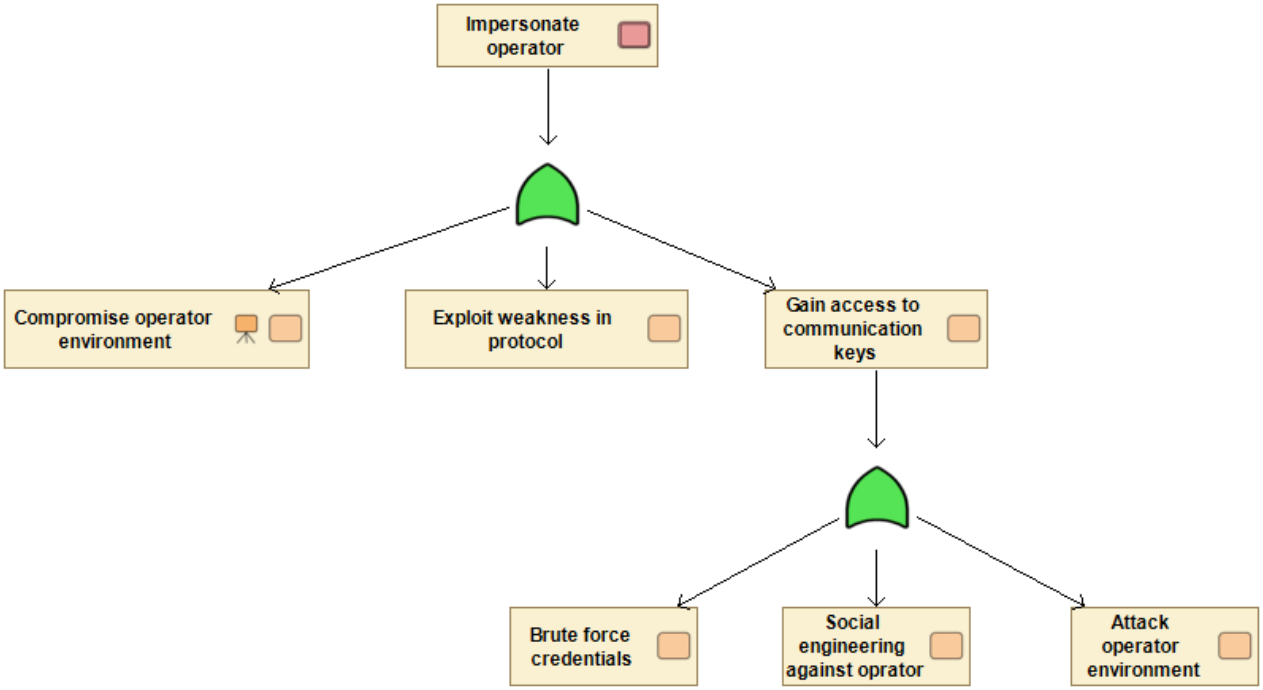

Figure 30 - Impersonate operator

A significantly more dangerous attack than simply impersonating another swarm member is to impersonate the operator itself. This assumes that there is a separation of privileges – and that there are certain, often dangerous operations that can only be performed by the operator. If a weakness in the protocol is found, no interaction with the real operator is necessary – otherwise, the attacker has to first compromise the operator environment (see Figure 30).

If successful, very little remains off the table for the attacker – the attack certainly implies a successful 0-day, as well as a total compromise of the availability and integrity of primary assets in the Service and Environment category (PA5, PA6, PA7, PA8, PA9 and PA10). Secondary assets are similarly affected.

7.4.8 Modify firmware

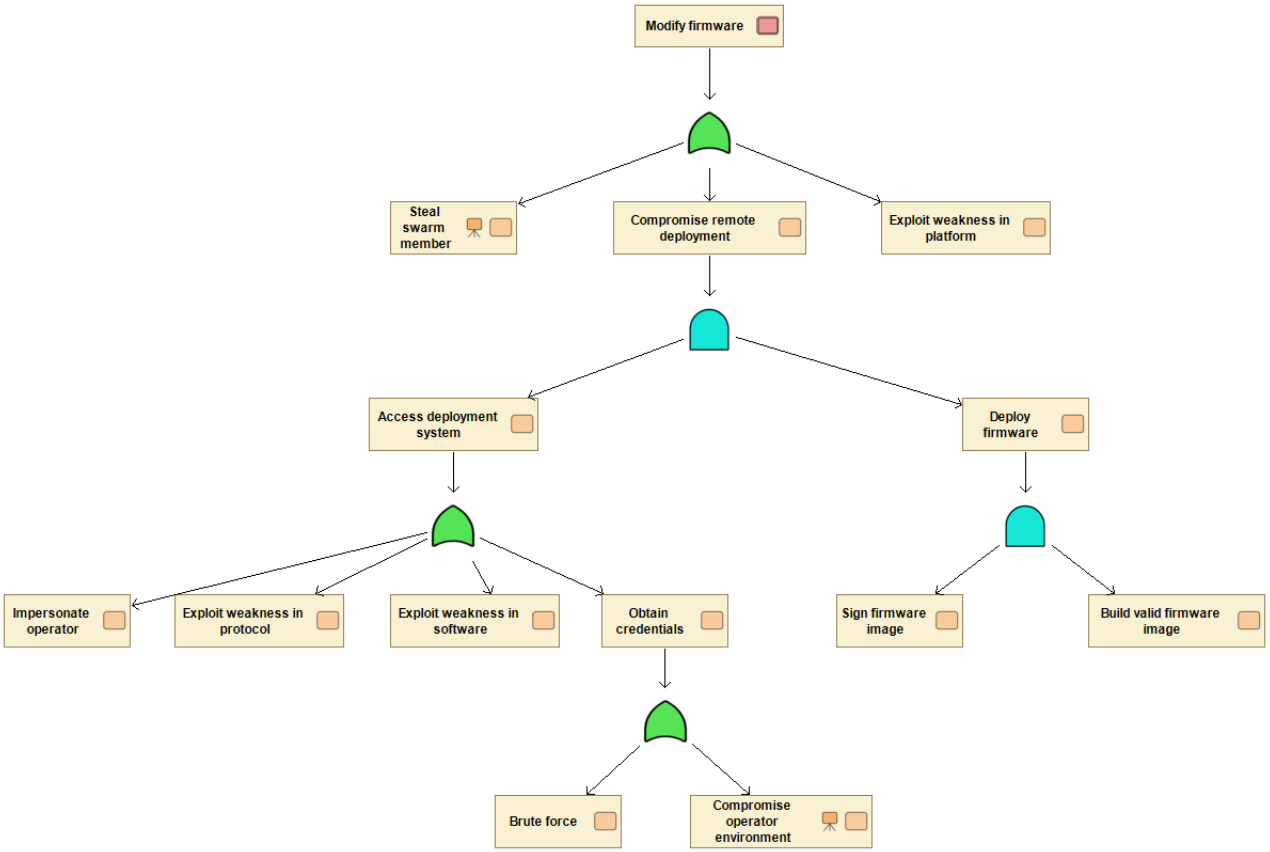

Figure 31 - Modify firmware

Like traditional computing systems, swarm members are also subject to software modifications resulting from physical access or from exploits affecting the underlying operating system. In our model, however, swarm members are also subject to software updates through a remote deployment system. As it is standard practice with software updates, the privilege of installing updates and the privilege of authorizing the update and certifying its authenticity is separated – the attacker has to compromise the deployment system and then deploy an integrity protected update that the system recognizes as valid (see Figure 31).

Modifying the firmware implies a total takeover of one swarm member – or if the attack is repeatable, most likely all members. Its effects can be similar to that of 7.4.6 or even 7.4.7, depending on the scope of the changes.

7.4.9 Compromise operator environment

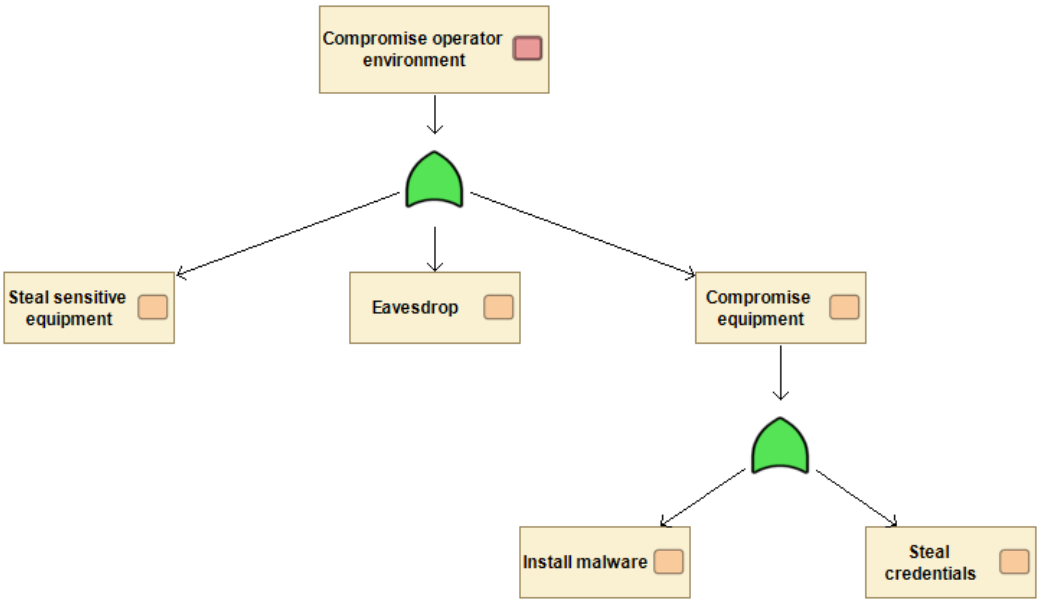

Figure 32 - Compromise operator environment

The weakest link in most systems is the human – and for our swarm, this relates to the operator. The operator environment includes the computers, software and network connection used by the operator, as well as any information stored on its premises. By compromising this environment, the attacker can gain access to the system without the need to exploit any weaknesses in the swarm itself (see Figure 32).

As this is a supporting attack, its effects on assets depend on the information obtained and on how that information is used. In the worst case, it can lead to a total takeover as in 7.4.7.

7.5 Use case I. (swarm of drones) attacks

7.5.1 Attacker motivations

The attacker motivated in diverting or sabotaging a rescue mission is possibly trying to cause more damage and desperation, additionally to a previously planned attack, to amplify the effect.

An attacker can be also motivated to steal a drone and sell or reprogram it for his purposes.

7.5.2 Methods of compromise

Spoofing (GPS, UWB) and jamming (Wifi, 3G/LTE) are the most obvious techniques to sabotage mission and redirect drones by preventing communication among swarm members and mission control (via WiFi or 3G/LTE) or to disturb localization (GPS/UWB) to cause emergency landing (drone) and mission abort (drone and rover). Bruteforcing WiFi can be used to replay messages and/or overload the channel.

ID	Attacks	Relevance to assets of the use case
A1	Brute force credentials	WiFi password, Communication library
A2	Exploit weakness in protocol	Communication library, MQTT
A3	Jam communication channel	GPS, UWB, WiFi, 3G/4G,
A4	Spoof authorized stop	Communication library
A5	Exploit weakness in platform	ROS

A6	Sign firmware image	Deployment tool
A7	Impersonate target	Communication library, MQTT
A8	Predict path	Communication library
A9	Exhaust consumables	Battery
A10	Spoof sensor data	GPS, UWB, Camera, Ultrasound

Table 7 - Methods of compromise for use case I.

7.6 Use case II. (automotive) attacks

7.6.1 Attacker motivations

The attacker can be motivated in diverting or sabotaging a delivery mission to cause financial loss to the delivery company or the recipient by delaying delivery. Another possible attacker motivation can be stealing of the car and transported goods.

A terrorist might be motivated to redirect self-driving cars to cause high traffic, or hit pedestrians, cause damage in the environment, the vehicle and people.

7.6.2 Methods of compromise

Spoofing (GPS, Ultrasound, LIDAR, Camera) and jamming (GPS, Wifi, 3G/LTE) are the most obvious techniques to sabotage mission and redirect vehicles by preventing communication among platooning members and mission control (via 3G/LTE) or to disturb localization (GPS) to cause emergency stop. Bruteforcing WiFi can be used to replay messages and/or overload the channel. The availability of the communication channel (TTEthernet deterministic wireless) with low latency is critical in this scenario. Jamming ultrasound with by high-pitch sounds or pressurized air requires carrying out the attack from a shorter (a few meters) distance compared to previous methods (from tens and a few hundreds of meters).

ID	Attacks	Relevance to use-case
A1	Brute force credentials	WiFi password
A2	Exploit weakness in protocol	Communication library, TTEthernet
A3	Jam communication channel	GPS, WiFi, 3G/4G
A4	Spoof authorized stop	Communication library
A5	Exploit weakness in platform	ROS, FogNode
A6	Sign firmware image	Deployment tool
A7	Impersonate target	Communication library, TTEthernet
A8	Predict path	Communication library, TTEthernet
A9	Exhaust consumables	Gas, Oil, Coolant
A10	Spoof sensor data	GPS, Camera, Ultrasound, LIDAR

Table 8 - Methods of compromise for use case II.

7.7 Use case III. (Logistics) attacks

7.7.1 Attacker motivations

An attacker may be interested to gain knowledge about the warehouse layout and locations of items of interests (goods and security personnel and surveillance system) or even hijack a drone to plan and implement a high-value theft.

7.7.2 Methods of compromise

Spoofing (Camera with QR codes, cloaking, LIDAR with lasers) and jamming (Wifi, Bluetooth, ZigBee all require jamming the 2,4 GHz ISM (Industrial Scientific Mediacl) band) are the most obvious techniques to sabotage mission and redirect rovers by preventing communication among swarm members and mission control (via WiFi or Bluetooth to cause mission abort (drone and rover). Bruteforcing (WiFi, ZigBee, Bluetooth) can be used to replay messages and/or overload the channel. Bluetooth manual control is another attack surface to be exploited by attackers to manually control the movement of the robots.

ID	Attacks	Relevance to use-case
A1	Brute force credentials	WiFi password, Bluetooth, Zigbee
A2	Exploit weakness in protocol	Communication library, Zigbee, MQTT, Bluetooth
A3	Jam communication channel	GPS, UWB, WiFi, 3G/4G, Bluetooth
A4	Spoof authorized stop	Yes
A5	Exploit weakness in platform	ROS
A6	Sign firmware image	Deployment tool
A7	Impersonate target	Communication library
A8	Predict path	Communication library
A9	Exhaust consumables	Battery
A10	Spoof sensor data	Camera, Laser

Table 9 - Methods of compromise for use case III.

7.8 Summary of use-case attack trees

As there are numerous overlaps among the use cases, the attack trees hereby presented contain notes about which use cases they apply.

Exploiting a possible weakness in the protocols used by the scenarios is summarized in the attack tree below (see Figure 33). The CPSwarm Communication library is used by all use cases.

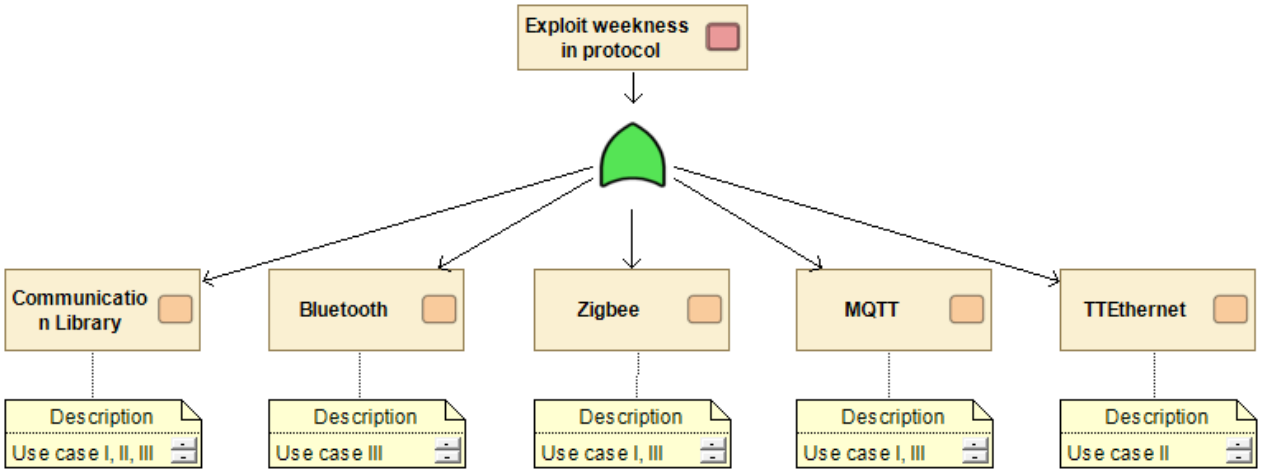

Figure 33 - Exploit weakness in protocol

The possibilities of modifying the environment conditions of the scenarios are summarized in the attack tree below (see Figure 34). Ultrasound sensors are employed by all use cases.

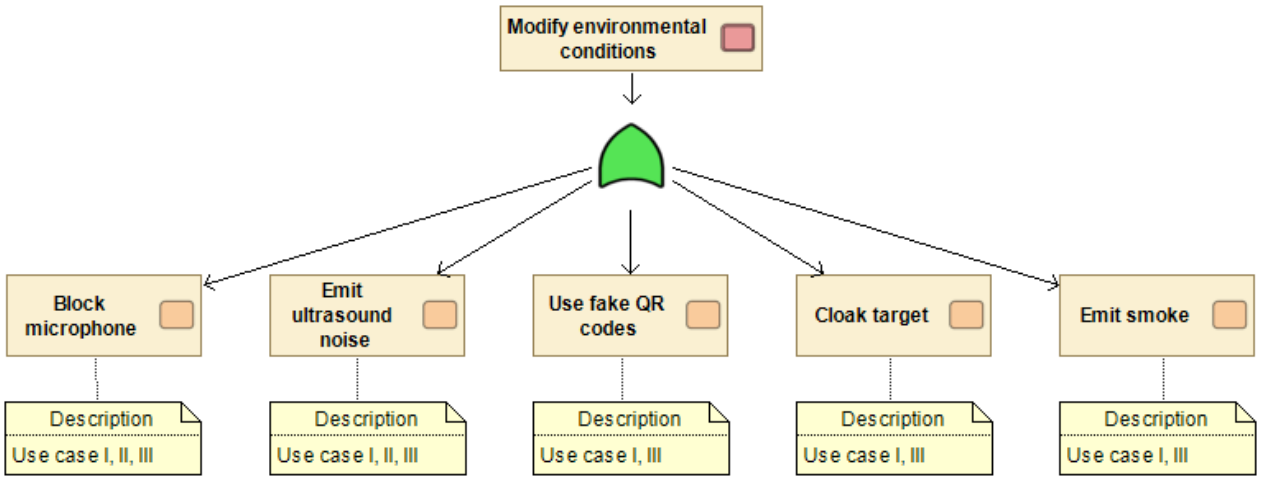

Figure 34 - Modify environmental conditions

Spoofing sensor by replaying messages, sound recordings, displaying fake QR codes, etc. in the scenarios are summarized in the attack tree above (see Figure 34). The relevance of technologies used in the use cases for spoofing is summarized below (see Figure 35).

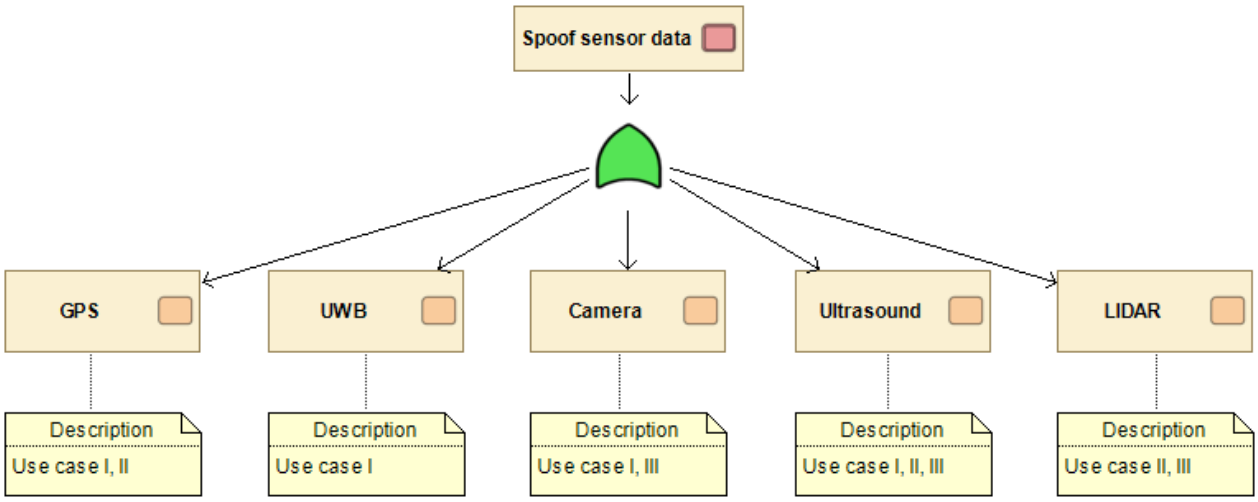

Figure 35 - Spoof sensor data

A denial-of service attack on communication technologies used in the scenarios is summarized in the attack tree below (see Figure 36). WiFi is used by all use cases.

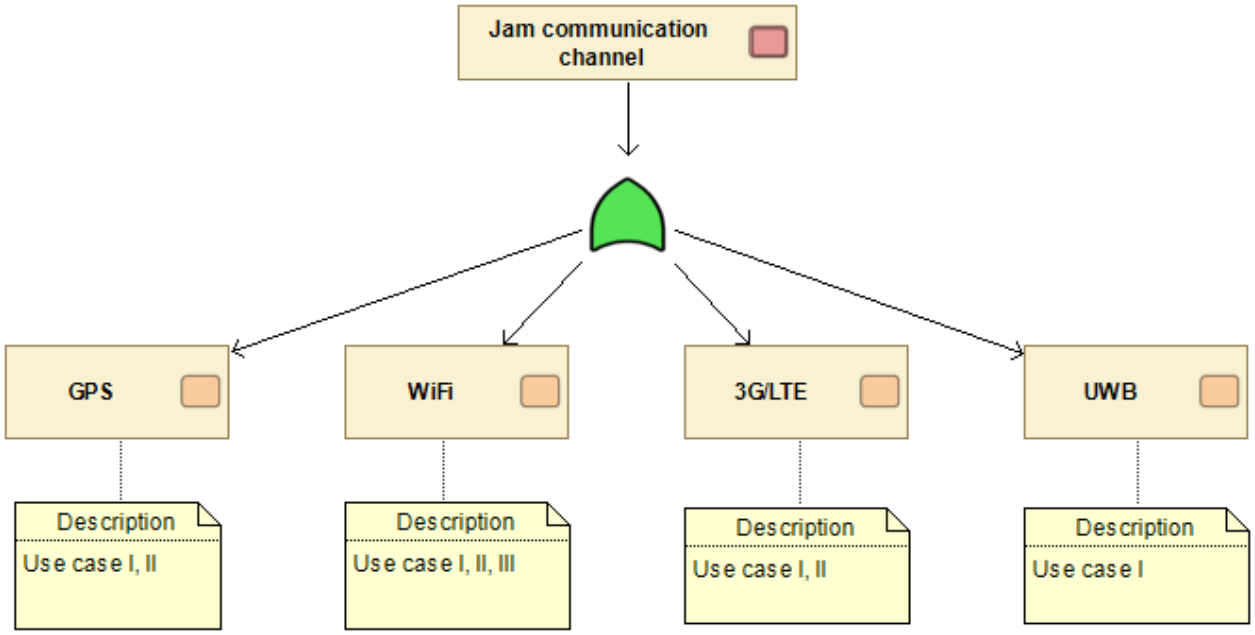

Figure 36 - Jam communication channel

8 Countermeasures

8.1 A study on attacks and their mitigations

In Chapter 7.2, the authors have identified attacker motivations and then lower level attacks that can corrupt assets corresponding to a swarm in mission. In Table 10 the low level attacks that can be mitigated have been collected. The authors have analysed whether they are in scope of the CPSwarm project – for example, physical security, social engineering and generic information security does not concern the main goal of the project and thus countermeasures or analyses regarding these have not been included. Some attacks can be carried out in many different ways depending on the application scenario and the type of CPSs used in the swarm, hence different countermeasures may apply to these.

ID	Attacks	In scope?	Category	Countermeasures
A1	Brute force credentials	Yes	Communication	See Chapter 8.2.3
A2	Exploit weakness in protocol	Yes	Communication	See Chapter 8.2.3
A3	Jam communication channel	No	Physical attack	N/A
A4	Spoof authorized stop	Yes	Communication	See Chapter 8.2.3
A5	Exploit weakness in platform	Yes	Hardening	See Chapter 8.2.1
A6	Sign firmware image	Yes	Deployment	See Chapter 8.2.2
A7	Impersonate target	Yes	Use case specific	See Chapter 8.4
A8	Predict path	Yes		
A9	Exhaust consumables	Yes		
A10	Spoof sensor data	No	Physical attacks	N/A
A11	Exploit weakness in software	No	Testing	
A12	Build valid firmware image	No	Obscurity	
A13	Gain physical access	No	Physical attack	
A14	Incapacitate swarm member	No		
A15	Modify environmental conditions	No		
A16	Steal sensitive equipment	No		
A17	Trigger component failure	No		
A18	Mislead operator	No	Social engineering	
A19	Install malware on operator environment	No	Generic information security	
A20	Steal operator credentials	No		

Table 10 - Summary of countermeasures

8.2 Design considerations and embedded countermeasures

The CPSwarm project is focused on building tools that aid the development of autonomous CPS swarms. The final goal of exploring the threat landscape from the perspective of future users and operators is both to help the Consortium to make better, more capable tools that have been designed with security in mind and to help these future users use these tools to build secure swarms.

While security analysis will have to be performed for each use case independently, unsurprisingly the attacks described so far point to a few major entry points:

- Hardware platform and firmware
- Communications infrastructure
- Operator environment
    - Deployment infrastructure
    - Monitoring Tool

A remote attacker has two choices: mount an attack against swarm members or mount an attack against the operator. The authors will not tackle the latter – it is the responsibility of the operator to properly secure its systems, both physically and from an IT security perspective. The former, however, is at the core of CPSwarm goals, and as such, the three relevant main areas where attacks are expected to happen need to be covered.

8.2.1 Hardware platform and firmware

The security of any software product depends heavily on the security of the underlying platform - no matter how carefully the developer builds the software, the system is likely to be compromised if the platform itself is vulnerable. The field of robotics has traditionally been a world of closed systems, with robots working without any network connection or only communicating on a local, closed network with their peers. Attacks against such systems often had to rely on the human element, like spreading malware through USB drives. As industries work toward increased connectivity, more and more devices are placed on public networks or networks where bridges exist to a public network – and as new attack surfaces open, the robotics industry now faces the challenge of securing these devices in the face of remote attackers.

The main robotics platform used in the CPSwarm project, Robot Operating System (ROS), is a Linux-based system with a number of custom packages and its own Inter-Process Communication (IPC) system. Its defaults are completely insecure – most stock firmware images contain no security features whatsoever, the ROS communication model[25] is devoid of any authentication or authorization scheme. While ROS2 is under development, and will eventually try to address some of these issues, it is not yet production ready and lacks the software and hardware ecosystem that was built around its predecessor. As such, if ROS-based devices are connected to the internet, they might be vulnerable to a variety of attacks (replay, impersonation, eavesdropping) even if no additional custom software is being used.

To set up the hardware and software platform, the following generic steps need to be taken:

1. First, **the platform needs to be updated**. For production environments – with the CPS having planned lifetimes measured in years – this includes using software that will be supported with security updates for the foreseeable future.
2. The second step is to configure the platform correctly and to **use only features that are inherently secure or not security sensitive**. What constitutes a correct configuration depends on the use case.
3. The third step is related to how the system is customized for the underlying hardware. **Sensors and actuators need to have proper error handling, input and output validation** – including sanity

[25] http://wiki.ros.org/ROS/Patterns/Communication

checks. Any communication equipment used must be set up to take advantage of low level security features provided by the physical layer.

4. Lastly, **settings need to be validated and saved** – to have a known starting configuration the system can be restored to. Follow up work also includes tracking and installing security updates.

These setup and maintenance steps are required to provide a stable platform for both swarm-related and other activities, but they are especially important for connected swarm applications where attack surfaces are significantly larger.

Deliverable D7.2 contains details on how to configure ROS to provide a more secure environment by using Linux configuration best practices, e.g. limiting the enabled services in ROS, do not run everything as root, etc. Following these practices limits the attack surface of the underlying operating system significantly.

8.2.2 Deployment infrastructure

In this context, the deployment infrastructure consists of all components supporting the remote installation and update of software components and configuration. This includes the Deployment Tool (see *D7.4 - Final Bulk deployment tool*) as developed in the CPSwarm project, but might also include other third party software components used by the operator. Since the basic premise of these components is that they allow the operator to change the software running on the CPS, if this system is compromised, the attacker can, in the worst case, compromise the swarm completely.

Since deployment is performed remotely, the **communications infrastructure is also involved**, and all the remarks and countermeasures described there also apply. Whether the deployment infrastructure should share the authentication scheme used by other communications should be evaluated on a case-by-case basis, but from a security perspective, a complete separation is a better option. In any case, authentication and authorization should only be the first line of defence.

Another layer of protection can be applied to the deployed artefacts themselves – by **signing all packages and validating their signature** before deployment on the device, even if an attacker can get through whatever authentication measures are in place, he won't be able to deploy arbitrary packages without also compromising the private key used by the operator to issue the signed packages. This also allows a separation of privileges between the personnel performing the deployment and the people responsible for the development and the issuance of valid software packages – in certain commercial applications, these groups might belong to different departments or even companies.

Even if the attacker gains access to both the deployment infrastructure itself and can produce a properly signed package, one last line of defence can be established by limiting the scope and privileges of the packages being deployed. While for simple applications, the package might be a full-fledged firmware image, in which case its compromise would lead to a total takeover, for most high complexity system, the deployment of new behaviour would be limited to the deployment of binaries and configuration files. In such a case, **industry standard isolation techniques** – limiting capabilities and containerization, file system isolation and so on – can be used to limit the damage that can be done by any deployed package. If this isolation is established correctly, the operator would still be able to shut down the rogue CPS, preventing further damage. Any such isolation must also include elements that limit behaviour to a safe range, checking the sanity of the input that is received from whatever control algorithm is deployed on the CPS. Ideally, any safety critical functionality would be protected by the isolation.

8.2.3 Communications infrastructure

While it is possible to use swarm algorithms that do not require direct communication between members (instead relying on sensory input), even such applications will likely communicate with the operator or the environment. This connectedness – as already mentioned in 8.2.1 – is the root of many security problems that plague modern robotics.

At the very least, any communication scheme utilized either must implement some form of **authentication** to limit participation in swarm communications. An alternative would be to limit the scope of communication

and the actions that could be remotely performed using these facilities – but even that would open up a significant attack surface.

The authentication scheme developed must be combined with the **strong integrity protection of messages**, so that for any message the recipient can determine whether it was sent by another member of the swarm (or the operator). In most use cases, it is also important to determine which swam member sent a particular message, and to be able to revoke access from compromised swarm members. This can be achieved using public key cryptography and by maintaining a chain of trust rooted at the operator. Provisioning devices at the time of their first use with certificates and establishing the trust relationship between the swarm and its new member is an important first step that must be performed in a secure environment, since before that trust relationship is established, no secure remote communications can take place.

Building on a working authentication scheme, combined with the strong integrity protection of messages, one can extend the scheme to cover **authorization**. In the context of swarm intelligences, this ensures that the operator is in a privileged position to issue certain commands that other parties in the communication – ordinary swarm members, IoT devices – cannot issue.

Authorization alone is insufficient to limit access to sensitive data, as once an authorized party requests such information others may eavesdrop – the solution is to protect the confidentiality of the messages using **encryption**. The need for encryption is use case dependent, and in certain cases might prove to be problematic – especially where performance and latency requirements make it impossible to use. Selective encryption and prioritization of messages is a possible solution – confidential telemetry data can usually tolerate higher latencies and jitter than high priority, safety critical messages (which might have no confidentiality requirement at all).

Wherever possible, industry standard technologies should be used – including for the physical layer of communications, which might bear some of the burden these countermeasures place on the implementer. Proving the correctness of any security relevant protocol is no easy task, and any concrete implementation of a protocol with these features will also be subject to attacks against the implementation itself.

The CPSwarm Communication library[26] aims to provide a secure layer for data exchange among swarm members, monitoring and deployment. More detailed description of the Communication Library is available in D7.2.

The security functionalities are to be provided by libsodium[27] (based on NaC[28]l). Libsodium is a popular solution for crypto library used by e.g.: WordPress, Discord, Secrets, Remembear. All cryptographic functions are based on:

- Edwards-Curve Digital Signature Algorithm (EdDSA)
- Encryption: XSalsa20 stream cipher
- Authentication: Poly1305 MAC

The following security dimensions will be addressed:

- Deployment tool is able to securely provide new node members (by generating their keys and signing their certificates)
- Access control is provided for provisioned nodes by certificate checking, using a pre-shared signing key
- Authentication is provided by signature checking

[26] https://github.com/cpswarm/swarmio
[27] https://github.com/jedisct1/libsodium
[28] https://nacl.cr.yp.to/

- Non-repudiation is provided by signature and timestamp checking for each packet
- Confidentiality is provided end-to-end by payload encryption
- Integrity checking is provided by using a tag for packet integrity
- Availability is maintained using each nodes security table, which stores valid authentication credentials.

Utilizing the security features (proven cryptographic functions and libraries) of the Communication Library the attack surface of network attacks shrinks to a level which is compliant with current best practices.

8.3 Summary of proposed general countermeasures

Any real world application of swarms should, at the very least:

- Build on up-to-date, supported, correctly configured platforms
- Implement security and safety critical functionality isolated from the main behaviour
- Use authenticated communications facilities, with dangerous actions requiring authorization
- Protect the confidentiality of sensitive communications with strong cryptography
- Implement remote deployment with multiple layers of security and isolation, or not at all
- Evaluate the impact of security features on the performance of the swarm

Within the CPSwarm project, a platform is being built that enables and empowers developers to achieve these goals and more. It is not on this project alone to supply all pieces of the puzzle – but the pieces the project supplies must help and not hinder these goals.

Table 11 presents the way in which the proposed general countermeasures fulfil the requirements presented in Chapter 2.5.,

ID	Requirement	Related attacks	Fulfillment
CRD-143	Passwords shall never be viewable at the point of entry or at any other time.	A20 - Steal operator credentials	Workbench, Monitoring tool, Deployment tool implementation
CRD-133	The system shall not be shut down for maintenance more than once in a 24-hour period.	A2 - Exploit weakness in protocol	Simple replay DoS protection by Communication Library
CRD-128	The system shall be protected against cyber attacks	A1 - Brute force credentials A2 - Exploit weakness in protocol A3 - Jam communication channel A4 - Spoof authorized stop A5 - Exploit weakness in platform A6 - Sign firmware image A19 - Install malware on operator environment	Simple replay DoS protection by Communication Library, ROS and operator environment hardening

ID	Requirement	Related attacks	Fulfillment
CRD-127	Attempts at accessing sensitive data by unauthorised users must be logged	A2 - Exploit weakness in protocol A6 - Sign firmware image A20 - Steal operator credentials	Workbench, Monitoring tool, Deployment tool implementation
CRD-126	Accessing sensitive data must be logged (User ID, Timestamp, etc.)	A2 - Exploit weakness in protocol A6 - Sign firmware image A20 - Steal operator credentials	Workbench, Monitoring tool, Deployment tool implementation
CRD-123	The solution should be in compliance with GDPR as well as national policies	A2 - Exploit weakness in protocol A6 - Sign firmware image A20 - Steal operator credentials	Workbench, Monitoring tool, Deployment tool implementation
CRD-119	Data processing and management must comply with relevant regulations	A2 - Exploit weakness in protocol A6 - Sign firmware image A20 - Steal operator credentials	Workbench, Monitoring tool, Deployment tool implementation
CRD-81	Software components running on the CPS shall be started with the lowest possible privileges.	A5 - Exploit weakness in platform	ROS hardening
CRD-78	The Deployment Agent shall use the list of trusted certificates supplied when the device is first provisioned to validate signatures.	A6 - Sign firmware image	Deployment tool implementation
CRD-76	The Deployment Manager shall provide a way to generate, import and export operator specific keys for code signatures.	A6 - Sign firmware image	Deployment tool implementation
CRD-75	The Deployment Agent shall verify the signatures of packages on boot and when updates are received.	A6 - Sign firmware image	Deployment tool implementation
CRD-73	The Deployment Tool shall implement secure over-the-air update functionality.	A6 - Sign firmware image	Deployment tool implementation
CRD-72	The Deployment Manager shall sign all packages with an operator specific key.	A6 - Sign firmware image	Deployment tool implementation

ID	Requirement	Related attacks	Fulfillment
CRD-68	All communications between swarm members shall be authenticated and integrity protected, with a per-message policy on encryption.	A1 - Brute force credentials A2 - Exploit weakness in protocol A4 - Spoof authorized stop	Secure Communication Library provides authentication, integrity protection and encryption.
CRD-67	All communications between the swarm and the tools in the workbench shall be authenticated, integrity protected and encrypted.	A1 - Brute force credentials A2 - Exploit weakness in protocol A4 - Spoof authorized stop	Secure Communication Library provides authentication, integrity protection and encryption.
CRD-64	The Code Generator and all the code generated shall be compliant to ISO 26262.	A13 - Gain physical access A14 - Incapacitate swarm member A15 - Modify environmental conditions A16 - Steal sensitive equipment A17 - Trigger component failure	Code Generator implementation
CRD-60	The communication between the Deployment Agent running on swarm members and the Deployment Manager shall be authenticated, authorized, encrypted, and integrity checked.	A1 - Brute force credentials A2 - Exploit weakness in protocol A4 - Spoof authorized stop	Secure Communication Library provides authentication, integrity protection and encryption.
CRD-35	The communication link between the swarm and the Monitoring Tool shall be authenticated and encrypted	A1 - Brute force credentials A2 - Exploit weakness in protocol A4 - Spoof authorized stop	Secure Communication Library provides authentication, integrity protection and encryption.

Table 11 - Countermeasures for requirements

8.4 Use case attacks and their mitigations

This section, in Table 12, lists the common mitigation solutions for the in-scope attacks related to the use cases.

ID	Attacks	Mitigation in use-case
A1	Brute force credentials	Changing cryptographic keys in every mission, certificate revocation makes brute-forcing much harder for the keys and certificates used by the Communication Library.
A2	Exploit weakness in protocol	The Communication Library uses approved cryptographic

		primitives and implementations. Use WPA2 in WiFi with strong pre-shared key or certificates. Use VPN over communication channel for Monitoring.
A3	Jam communication channel	The communication layer of the operating system may find another route/channel or interface (if available), which is not jammed to resume communication. Devices detecting jamming can leave the jammed area and report to others.
A4	Spoof authorized stop	Communication Library prohibits message replay and modification by employing integrity checking and message counting. (Details available in D7.2)
A5	Exploit weakness in platform	ROS hardening lowers the attack surface of the underlying Linux systems used by the swarm members.
A6	Sign firmware image	Deployment tool signs the firmware image, which the swarm members can check.
A7	Impersonate target	The Communication Library bases identity protection on private keys and certificate revocation (a missing member can be blacklisted).
A8	Predict path	Communication Library prohibits message eavesdropping by encrypting traffic among swarm members and monitoring.
A9	Exhaust consumables	Simple replay-based DoS attack mitigation in Secure Communication Library is implemented
A10	Spoof sensor data	Integrity protection on transmitted data prevents data modification over the communication channel, but does not protect from physical access. Sensor fusion and anomaly detection can be utilized to recognise fake data, but it is out of the scope of the project.
A11	Exploit weakness in software	Operators should not enable Internet access on Monitoring and Deployment server and swarm members to lower virus/malware infection risks. Software hardening lowers the attack surface as well.

Table 12 - Use case countermeasures

9 Risk assessment

9.1 Methodology

A risk assessment is the combination of the following two procedures:

1. Identifying and analysing potential events that may negatively impact assets and
2. Evaluating the risk: making judgements on the tolerability of the identified risks while considering the influencing factors.

In short, a risk assessment analyses what can go wrong, how likely it is to happen, what the potential consequences are and how tolerable the risk is.

In Chapter 5, the authors have already identified the assets that need to be protected, while Chapter 7 has already described potential events and attacks in sections 7.2 and 7.4 respectively, by using attack trees. What remains to complete the risk assessment is the definitions of the metrics based on which it can be determined the likelihood and severity of threats to identify risks. When evaluating the likelihood, the authors will be using a qualitative approach. The likelihood scale and the interpretation for the levels are presented in Table 13.

Likelihood	Qualitative interpretation
3: Certain	There is a high chance that the scenario successfully occurs in a short time
2: Likely	There is a high chance that the scenario successfully occurs during the life time of the application of the swarm
1: Unlikely	There is little or no chance that the scenario successfully occurs in a short time

Table 13 - Likelihood evaluation

To determine the severity of an attack, the following three levels are used:

- *Low* (1): Indirect or negligible attacks on the swarm fall into this category. The attacker can also obtain access to information, which may help executing other attacks against the swarm or the operators.
- *Medium* (2): The attacker can access sensitive information, data collected by swarm members, mission related parameters; or can cause persistent delays in the mission. The confidentiality and/or integrity of swarm data is endangered by the attacker.
- *High* (3): The attacker can obtain control of the swarm, or can cause permanent damage to the swarm, humans or the environment.

On Table 14, the risk levels are estimated from the likelihood of attacks and their severity in a risk matrix. The risk value can take the following levels:

- *High*: the threat significantly endangers related assets
- *Medium*: the threat has a noticeable effect on the security of the related assets
- *Low*: The threat has a minor effect on the security of the related assets

To make the following table better readable, the authors assign different colours to different risks: high –red, medium – yellow and low – green. For instance, a likely accident with high (3) severity has a high level risk.

	Severity		
Likelihood	1	2	3
3: Certain	Medium	High	High
2: Likely	Low	Medium	High
1: Unlikely	Low	Low	Medium

Table 14 - Risk matrix

Now that all the metrics have been set to evaluate the risks, in the following sections the authors determine the severity and likelihood of the low-level attacks described in Section 7.4t. Then the risk level of the attack scenarios presented in Section 7.2 is calculated, based on the set of sufficient attacks to realize them. First the authors conduct a risk assessment without any countermeasures considered in Section 9.1.1 and then, in Section 9.1.3, the severity and likelihood of attacks is recalculated, when the proposed countermeasures from Section 8 are applied, and finally determine the risk levels with countermeasures.

9.1.1 Attacking a swarm member vs. attacking the swarm

The application of swarms of smaller robots, with limited capabilities comes from the idea that these robust and failure tolerant systems can be used in safety-critical missions, where the failure of a fraction of the swarm members does not have a severe impact on the whole swarm's mission. However, when the size of the swarm is not measured in tens or hundreds of robots, even attacking a single member can affect the mission.

Having a limited number of swarm members does not mean that the mission will be less efficient, since their behaviour is optimized according to the size of the swarm, and in lots of cases when there is a scarcity of operational space, it is better to use smaller swarms. In the case of swarm with a handful of members, the trade-off is endangering the success of the mission by one or several members' failure or misbehaviour, either caused by adversaries or other, natural causes like hardware faults, software bugs or environmental conditions.

The consequences of attacking a single member of the swarm could include

- Reduced efficiency in executing the mission – since the behaviour, distribution, etc. is optimized for a fixed number of members;
- Physical harm to other swarm members due to collisions;
- Change in the expected swarm behaviour - in some configurations swarm members could trigger events which could result in switching to other behaviours;
- When there is a clearly established hierarchy among swarm members, attacking the master members could severely abuse the mission.

The last point brings the authors to the second option, to perform an attack against the whole swarm. Hijacking the leading member (if present) is on the border of these categories since it could result in an attack against the whole swarm. Of course, attacking the whole swarm is a very clear way to stop its mission or delay it for a sufficiently long time. However, it may require less effort to just disable one or few members to mislead or completely disrupt the swarm.

If an attack against one swarm member, once successful, can be performed again and again against the rest of the swarm members with little additional effort, the severity of the attack increases significantly. Likelihoods are not affected.

9.1.2 Risk assessment

First, for each low-level attack (presented in Section 7.4) the authors determine their likelihood and severity, so it is possible to calculate the risk related to them as seen in Table 15 below. The calculations are presented without considering countermeasures.

ID	Attack	Likelihood	Severity	Risk Level
A1	Brute force credentials	Unlikely	2	Low
A2	Exploit weakness in protocol	Likely	2	Medium
A3	Jam communication channel	Likely	2	Medium
A4	Spoof authorized stop	Likely	3	High
A5	Exploit weakness in platform	Likely	2	Medium
A6	Sign firmware image	Unlikely	2	Low
A7	Impersonate target	Likely	1	Low
A8	Predict path	Likely	2	Medium
A9	Exhaust consumables	Likely	2	Medium
A10	Spoof sensor data	Likely	2	Medium
A11	Exploit weakness in software	Likely	2	Medium
A12	Build valid firmware image	Unlikely	3	Medium
A13	Gain physical access	Unlikely	3	Medium
A14	Incapacitate swarm member	Likely	2	Medium
A15	Modify environmental conditions	Likely	2	Medium
A16	Steal sensitive equipment	Unlikely	3	Medium
A17	Trigger component failure	Unlikely	2	Low
A18	Mislead operator	Unlikely	1	Low
A19	Install malware on operator environment	Likely	1	Low
A20	Steal operator credentials	Likely	2	Medium

Table 15 - Risk assessment of attacks

From the underlying attacks, it is possible to estimate the overall risk level of the threat scenarios by choosing the highest risk available from the risks identified for basic attacks above, as presented in Table 16.

Attacker goal	Related attacks	Overall risk
Sabotage mission	A1, A2, A3, A4, A5, A6, A9, A11, A12, A14, A15, A16, A17, A18, A19, A20	High
Cause financial damage to swarm operator	A1, A2, A3, A4, A5, A6, A8, A9, A10, A11, A12, A13, A14, A15, A16, A17, A18, A19, A20	High
Cause physical harm to human being or property	A1, A2, A4, A5, A6, A10, A11, A14, A15, A16, A17, A18, A19 A20	High
Disturb environment or bystanders	A1, A2, A4, A5, A6, A10, A11, A14, A15, A16, A17, A18, A19 A20	High
Steal swarm member	A2, A4, A5, A6, A10, A11, A14, A15, A16, A17 A18, A19, A20	High
Steal sensitive data	A1, A2, A5, A6, A11, A15, A18, A19, A20	Medium

Table 16 - Attacker goals, related attacks and their overall risk

As it can be seen in the table, five from the total six attacker goals identified have a high risk, since the attack, which has a high risk (A4) can help realize them. In the next section, the authors redo the risk assessment by taking the proposed countermeasures into account.

9.1.3 Risk assessment with countermeasures

First, the authors look at whether a low-level attack can be mitigated and if so, with what type of countermeasure from the ones described in Section 7. Then the likelihoods and hence the risk level of these attacks is recalculated. Results are presented in Table 17.

ID	Attack	Countermeasure	Likelihood	Severity	Risk Level
A1	Brute force credentials	Communication	Unlikely	2	Low
A2	Exploit weakness in protocol	Communication	Unlikely	2	Low
A3	Jam communication channel	Communication	Unlikely	2	Low
A4	Spoof authorized stop	Communication	Unlikely	3	Medium
A5	Exploit weakness in platform	Hardening	Unlikely	2	Low
A6	Sign firmware image	Deployment	Unlikely	2	Low
A7	Impersonate target	Communication	Unlikely	1	Low
A8	Predict path	Communication	Unlikely	2	Low
A9	Exhaust consumables	Communication	Unlikely	2	Low
A10	Spoof sensor data	Communication	Unlikely	2	Low

ID	Attack	Countermeasure	Likelihood	Severity	Risk Level
A11	Exploit weakness in software	Secure coding and review	Unlikely	1	Low
A12	Build valid firmware image	Out of scope	Unlikely	3	Medium
A13	Gain physical access	Out of scope	Unlikely	3	Medium
A14	Incapacitate swarm member	Out of scope	Likely	2	Medium
A15	Modify environmental conditions	Out of scope	Likely	2	Medium
A16	Steal sensitive equipment	Out of scope	Unlikely	3	Medium
A17	Trigger component failure	Out of scope	Unlikely	2	Low
A18	Mislead operator	Out of scope	Unlikely	1	Low
A19	Install malware on operator environment	Out of scope	Likely	1	Low
A20	Steal operator credentials	Out of scope	Likely	2	Medium

Table 17 - Attacks with recalculated risks after applying countermeasures

Now, the overall risks are recalculated for the attacker goals in Table 18.

Attacker goal	Related attacks	Overall risk
Sabotage mission	A1, A2, A3, A4, A5, A6, A9, A11, A12, A14, A15, A16, A17, A18, A19, A20	Medium
Cause financial damage to swarm operator	A1, A2, A3, A4, A5, A6, A8, A9, A10, A11, A12, A13, A14, A15, A16, A17, A18, A19, A20	Medium
Cause physical harm to human being or property	A1, A2, A4, A5, A6, A10, A11, A14, A15, A16, A17, A18, A19 A20	Medium
Disturb environment or bystanders	A1, A2, A4, A5, A6, A10, A11, A14, A15, A16, A17, A18, A19 A20	Medium
Steal swarm member	A2, A4, A5, A6, A10, A11, A14, A15, A16, A17 A18, A19, A20	Medium
Steal sensitive data	A1, A2, A5, A6, A11, A15, A18, A19, A20	Medium

Table 18 - Recalculated risk levels to attacker goals

As it can be seen in the table, all risks are now reduced to medium as a consequence of applying countermeasures and thus eliminating high risks.

9.2 Use case I (swarm of drones)

9.2.1 Design considerations, embedded countermeasures

Because of the scenario is located in an open space and possibly dangerous environment, it is difficult to tell whether a drone has been captured or merely fell victim of a natural event. In case the malicious actor infiltrated the network, and is successfully manipulating a member, it can influence other swarm members to fulfil his purposes.

Possible consequences of compromised security:

- Cause physical pain to human beings or property,
- Sabotage mission,
- Steal sensitive information (e.g. location of rescue target, access camera).

Threats and vulnerabilities:

- Eavesdrop on communication (e.g. listening to radio link),
- DOS attack on asset communication (e.g. radio jamming),
- GPS/UWB spoofing,
- Modify mission parameters (e.g. overwriting target information).

This use case's risk quantification is high, as security compromise can result in severe harm or even death of humans.

9.2.2 Risk assessment

Due to the possible consequences, the risk assessment resulted in elevated severity, therefore some of the attacks produced higher risk levels, compared to the generic observations, as indicated in Table 19.

ID	Attack	Likelihood	Severity	Risk Level
A1	Brute force credentials	Unlikely	2	Medium
A2	Exploit weakness in protocol	Likely	3	High
A3	Jam communication channel	Likely	2	Medium
A4	Spoof authorized stop	Likely	3	High
A5	Exploit weakness in platform	Likely	3	High
A6	Sign firmware image	Unlikely	2	Low
A7	Impersonate target	Likely	2	Medium
A8	Predict path	Likely	2	Medium
A9	Exhaust consumables	Likely	2	Medium
A10	Spoof sensor data	Likely	2	Medium
A11	Exploit weakness in software	Likely	2	Medium

Table 19 - Risk assessment for use case I.

9.2.3 Risk assessment with countermeasures

The introduction of the countermeasures described previously, lower the likelihood of many attacks, as the attacker encounters a smaller attack surface or should have sophisticated techniques and insider knowledge, as indicated in Table 20.

ID	Attack	Countermeasure	Likelihood	Severity	Risk Level
A1	Brute force credentials	Communication	Unlikely	2	Low
A2	Exploit weakness in protocol	Communication	Unlikely	3	Medium
A3	Jam communication channel	Communication	Unlikely	2	Low
A4	Spoof authorized stop	Communication	Unlikely	3	Medium
A5	Exploit weakness in platform	Hardening	Unlikely	3	Medium
A6	Sign firmware image	Deployment	Unlikely	2	Low
A7	Impersonate target	Communication	Unlikely	2	Low
A8	Predict path	Communication	Unlikely	2	Low
A9	Exhaust consumables	Communication	Unlikely	2	Low
A10	Spoof sensor data	Communication	Unlikely	2	Low
A11	Exploit weakness in software	Secure coding and review	Unlikely	2	Low

Table 20 - Risk assessment with countermeasures for use case I.

9.3 Use case II (automotive)

9.3.1 Design considerations, embedded countermeasures

Because of the scenario is located in an open space (public roads) and a truck has potentially significant amount of cargo and mass, the effect of malicious control can cause high-severity events.

Possible consequences of compromised security:

- Safety critical system failure (accident or crash)
- Theft of vehicles and transported goods
- Delay of delivery

Threats and vulnerabilities:

- Nonavailability (e.g. by DOS attack on leader-follower-link)
- Integrity loss of leader-follower communication (e.g. man-in-the-middle)

The system is operating in public roads. Trucks could be abused to create very dangerous situations. This use case has the highest risk level.

9.3.2 Risk assessment

Due to the possible consequences, the risk assessment resulted in elevated severity, therefore most of the attacks produced higher risk levels, compared to the generic observations, listed in Table 21.

ID	Attack	Likelihood	Severity	Risk Level
A1	Brute force credentials	Unlikely	3	Medium
A2	Exploit weakness in protocol	Likely	3	High
A3	Jam communication channel	Likely	3	High
A4	Spoof authorized stop	Likely	3	High
A5	Exploit weakness in platform	Likely	3	High
A6	Sign firmware image	Unlikely	3	Medium
A7	Impersonate target	Likely	2	Medium
A8	Predict path	Likely	2	High
A9	Exhaust consumables	Likely	3	High
A10	Spoof sensor data	Likely	3	High
A11	Exploit weakness in software	Likely	3	High

Table 21 - Risk assessment for use case II.

9.3.3 Risk assessment with countermeasures

The introduction of the countermeasures described previously lower the likelihood of many attacks, as the attacker encounters a smaller attack surface or should have sophisticated techniques and insider knowledge. The levels are indicated in Table 22.

ID	Attack	Countermeasure	Likelihood	Severity	Risk Level
A1	Brute force credentials	Communication	Unlikely	3	Medium
A2	Exploit weakness in protocol	Communication	Unlikely	3	Medium
A3	Jam communication channel	Communication	Unlikely	3	Medium
A4	Spoof authorized stop	Communication	Unlikely	3	Medium
A5	Exploit weakness in platform	Hardening	Unlikely	3	Medium
A6	Sign firmware image	Deployment	Unlikely	3	Medium
A7	Impersonate target	Communication	Unlikely	2	Medium
A8	Predict path	Communication	Unlikely	2	Medium

A9	Exhaust consumables	Communication	Unlikely	3	Medium
A10	Exploit weakness in software	Secure coding and review	Unlikely	3	Medium

Table 22 - Risk assessment with countermeasures for use case II.

9.4 Use case III (logistics)

9.4.1 Design considerations, embedded countermeasures

Because of the scenario is located in a closed space (factory/warehouse), which should have a relatively high safety and security requirement, the possible attack surface is restricted compared to other use cases. Secure entrance and security personnel along with a surveillance system in place prevent and detect malicious actors efficiently.

Possible consequences of compromised security:

- Theft of items
- System downtime (non-availability)
- Misdelivery of items (e.g. putting the wrong item into a shipping container)

Threats and vulnerabilities:

- DOS on Wi-Fi infrastructure
- Attack mission control (e.g. "man-in-the-middle")
- Attack on robot control (local Bluetooth)

Risk quantification of this use case is low, as it is located in a physically protected space and surrounded with a limited amount of humans.

9.4.2 Risk assessment

Due to the limited access, the risk assessment resulted in lowered likelihood, therefore some of the attacks produced higher lower levels, compared to the generic observations, as indicated in Table 23.

ID	Attack	Likelihood	Severity	Risk Level
A1	Brute force credentials	Unlikely	2	Low
A2	Exploit weakness in protocol	Unlikely	2	Low
A3	Jam communication channel	Likely	2	Medium
A4	Spoof authorized stop	Likely	2	Medium
A5	Exploit weakness in platform	Likely	2	Medium
A6	Sign firmware image	Unlikely	2	Low
A7	Impersonate target	Likely	1	Low
A8	Predict path	Likely	2	Medium
A9	Exhaust consumables	Unlikely	2	Low

A10	Spoof sensor data	Likely	2	Medium

Table 23 - Risk assessment for use case III.

9.4.3 Risk assessment with countermeasures

The introduction of the countermeasures described previously lower the likelihood of many attacks, as the attacker encounters a smaller attack surface or should have sophisticated techniques and insider knowledge. The levels are indicated in Table 24.

ID	Attack	Countermeasure	Likelihood	Severity	Risk Level
A1	Brute force credentials	Communication	Unlikely	2	Low
A2	Exploit weakness in protocol	Communication	Unlikely	2	Low
A3	Jam communication channel	Communication	Unlikely	2	Low
A4	Spoof authorized stop	Communication	Unlikely	2	Low
A5	Exploit weakness in platform	Hardening	Unlikely	2	Low
A6	Sign firmware image	Deployment	Unlikely	2	Low
A7	Impersonate target	Communication	Unlikely	1	Low
A8	Predict path	Communication	Unlikely	2	Low
A9	Exhaust consumables	Communication	Unlikely	2	Low
A10	Spoof sensor data	Secure coding and review	Unlikely	2	Low

Table 24 - Risk assessment with countermeasures for use case III.

10 Conclusion

This deliverable presented the state of the art assets, methodology, risk assessment, countermeasures for swarms of CPS, in particular for the use cases of the CPSwarm project as examples. Utilizing this document, the reader is able to look up relevant standards, gains inspiration from the use cases and can infer possible attacker motivations and attack surfaces for his own project. Following the methodology presented as guidance, from the discovery of assets, to suggesting countermeasures can be assessed for other systems. The presented tools, such as the Attack Tree Plugin for Modelio is available open source, as well as the CPSwarm Communication Library, which implements numerous security features necessary for secure interaction with and among swarm members.

Glossary of abbreviations

CIA	Confidentiality Integrity Availability
CPS	Cyber Physical Systems
CPU	Central Processor Unit
CVE	Common Vulnerabilities and Exposures
GDPR	*General Data Protection Regulation*
GPS	Global Positioning System
IEC	International Electrotechnical Commission
IOT	Internet of Things
IP	Internet Protocol
IR	Infrared
ISO	International Organization for Standardization
LIDAR	Light Detection and Ranging
MQTT	Message Que Telemetry Transport
ROS	Robot Operating System
SAR	Search and Rescue
TCP	Transmission Control Protocol
ToE	Target of Evaluation
UAV	Unmanned Aerial Vehicle
UWB	Ultra Wide Band

List of figures

List of tables

References

[1] "The STRIDE Threat Model," Microsoft, 11 12 2009. [Online]. Available: https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx. [Accessed 10 5 2018].

[2] [Online]. Available: https://en.wikipedia.org/wiki/DREAD_(risk_assessment_model).

[3] [Online]. Available: http://www.octotrike.org/.

[4] [Online]. Available: https://www.safaribooksonline.com/library/view/risk-centric-threat/9780470500965/c06.xhtml.

[5] A. ". e. a. Agarwal, “VAST Methodology: Visual, Agile, and Simple Threat Modeling.,” 2016.

[6] B. Schneier, “Attack trees,” *Dr. Dobb's Journal,* 1999.

[7] “The COSSIM H2020 project,” [Online]. Available: http://www.cossim.org.

[8] C. G. T. P. P. S. E. Schmittner, “Security application of failure mode and effect analysis (FMEA),” *Computer Safety, Reliability, and Security,* 2014.

[9] C. S. e. al., “Security Application of Failure Mode and Effect Analysis”.

[10] E. S. M. Ruijters, “Fault tree analysis,” *Comput. Sci. Rev. 15(C),* 2015.

[11] D. J. Marshall, “An Introduction to Fault Tree Analysis”.

[12] B. B. B. K. a. G. E. Ernő Jeges, “MEFORMA Security Evaluation Methodology,” 2014.

[13] B. Schneier, “https://www.schneier.com/academic/archives/1999/12/attack_trees.html,” 1999.

[14] [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:31985L0374&from=EN.

[15] [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32001L0095&from=EN.

[16] “The Robolaw project,” [Online]. Available: http://www.robolaw.eu/publicdocs.htm.

[17] [Online]. Available: http://www.beuc.eu/publications/beuc-x-2017-039_csc_review_of_product_liability_rules.pdf.

[18] [Online]. Available: https://ec.europa.eu/transport/sites/transport/files/com20160766_en.pdf.

[19] [Online]. Available: http://www.reersafety.com/pt/en/safety-guide/safety-in-the-working-environment/iec-62061-sil.