

Article

SESAME: Automated Security Assessment of Robots and Modern Multi-Robot Systems

Manos Papoutsakis ¹, George Hatzivasilis ^{2,*}, Emmanouil Michalodimitrakis ¹, Sotiris Ioannidis ², Maria Michael ³, Antonis Savva ³, Panagiota Nikolaou ⁴, Eftychia Stokkou ⁵ and Gizem Bozdemir ⁶

¹ Institute of Computer Science Foundation for Research and Technology—Hellas (FORTH), 70013 Heraklion, Greece; papoutsak@ics.forth.gr (M.P.); manmix@ics.forth.gr (E.M.)

² Department of Electrical and Computer Engineering, Kounoupidiana Campus, Technical University of Crete, 73100 Chania, Greece; sioannidis@tuc.gr

³ KIOS Center of Excellence, University of Cyprus, Nicosia 1678, Cyprus; mmichael@ucy.ac.cy (M.M.); savva.d.antonis@ucy.ac.cy (A.S.)

⁴ Department of Electrical and Computer Engineering, Cyprus Campus, University of Central Lancashire, Preston PR1 2HE, UK; pnikolaou1@uclan.ac.uk

⁵ Cyprus Civil Defence, Lefkosia 2404, Cyprus; estokkou@cd.moi.gov.cy

⁶ PAL Robotics SL, 08005 Barcelona, Spain; gizem.bozdemir@pal-robotics.com

* Correspondence: gchatzivasilis@tuc.gr

Abstract: As robotic systems become more integrated into our daily lives, there is growing concern about cybersecurity. Robots used in areas such as autonomous driving, surveillance, surgery, home assistance, and industrial automation can be vulnerable to cyber-attacks, which could have serious real-world consequences. Modern robotic systems face a unique set of threats due to their evolving characteristics. This paper outlines the SESAME project's methodology for the automated security analysis of multi-robot systems (MRS) and the production of Executable Digital Dependability Identities (EDDIs). Addressing security challenges in MRS involves overcoming complex factors such as increased connectivity, human–robot interactions, and a lack of risk awareness. The proposed methodology encompasses a detailed process, starting from system description and vulnerability identification and moving to the generation of attack trees and security EDDIs. The SESAME security methodology leverages structured repositories like Common Vulnerabilities and Exposures (CVE), Common Weakness Enumeration (CWE), and Common Attack Pattern Enumeration and Classification (CAPEC) to identify potential vulnerabilities and associated attacks. The introduction of Template Attack Trees facilitates modeling potential attacks, helping security experts develop effective mitigation strategies. This approach not only identifies, but also connects, specific vulnerabilities to possible exploits, thereby generating comprehensive security assessments. By merging safety and security assessments, this methodology ensures the overall dependability of MRS, providing a robust framework to mitigate cyber–physical threats.

Keywords: security evaluation; security assurance; robotic systems; attack trees; CVE; RVD



Academic Editors: Federico Rossi, Cinzia Bernardeschi and Gloria Gori

Received: 20 December 2024

Revised: 19 February 2025

Accepted: 20 February 2025

Published: 26 February 2025

Citation: Papoutsakis, M.; Hatzivasilis, G.; Michalodimitrakis, E.; Ioannidis, S.; Michael, M.; Savva, A.; Nikolaou, P.; Stokkou, E.; Bozdemir, G. SESAME: Automated Security Assessment of Robots and Modern Multi-Robot Systems. *Electronics* **2025**, *14*, 923. <https://doi.org/10.3390/electronics14050923>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The presence of software and hardware vulnerabilities in robotic systems presents a substantial risk with potentially dire consequences [1,2]. Exploiting these vulnerabilities can result in various damaging outcomes, such as financial losses, the exposure of sensitive information, the erosion of customer trust, damage to critical assets, and even human injuries or fatalities. Given the active role robotic systems play across multiple industry

sectors—including automotive, energy (both traditional and alternative), food, pharmaceuticals, aerospace, and more—these sectors all become potential targets for adversaries.

Prioritizing the security of robotic systems is therefore essential [3,4]. This responsibility should not rest solely with robot designers and operators; it must also involve standards creators, software developers, robot vendors, and security experts. The goal of these roles is to make exploiting robot vulnerabilities difficult and resource-intensive, thereby enhancing the overall security of robotic systems.

Modern robotic systems face a distinct set of threats due to their evolving nature [5–7]. These systems have become integral to daily life, embedded in applications such as cars, appliances, surveillance platforms, and medical equipment, often operating close to humans. Despite their prevalence, many of these systems lack built-in security mechanisms to guard against malicious threats. Additionally, they require connectivity to the external world for monitoring and maintenance, introducing new attack surfaces through APIs. Moreover, administrators often lack awareness of emerging risks, as traditional industrial robot environments were previously closed and considered secure. Consequently, conducting security assessments for robotic systems has become an essential yet challenging task.

The rest of the paper is organized as follows: Sector 2 reviews the background theory and the related works for security assessment. Section 3 presents the proposed solution for the security assessment of robotic systems of the EU-funded project SESAME. Section 4 gives the implementation details and Section 5 demonstrates the application of SESAME in two piloting environments, for critical infrastructure monitoring with drones and for healthcare operation with robotic assistants, respectively. Section 6 discusses scalability and implementation challenges. Finally, Section 7 concludes this work and provides directions for future works.

2. Background and Related Works

This section defines the problem of security assessment in robotic systems and provides an overview of related works.

2.1. The Challenges of Security Assessment

The security assessment of robotic systems is critical as these systems become increasingly integrated into various aspects of daily life, including autonomous driving, surveillance, surgery, home assistance, and industrial automation [8,9]. The complexity and connectivity of robotic systems introduce numerous vulnerabilities, making them potential targets for cyber-attacks [10,11]. This section outlines the challenges in securing robotic systems, discusses existing methodologies, and reviews related works in the field.

Robotic systems face significant cybersecurity challenges due to their integration with various technologies and their deployment in diverse environments [1]. The Robot Operating System (ROS) and its successor ROS2 exemplify these challenges, as follows:

- ROS and ROS2. ROS is a standardized middleware for robotics, facilitating communication among diverse robot clusters [3,4]. However, it has several vulnerabilities, such as plain-text communications and unprotected TCP ports. ROS2 addresses some of these issues by integrating the Data Distribution Service (DDS) standard for secure communication and implementing robust access control through ROS2 Security. Despite these improvements, continuous monitoring and adherence to security best practices remain essential;
- Industrial robots. The interconnectedness of industrial robots expands potential attack points [11–14]. Historically operated in isolated environments, these robots are now integrated into information and communication technology (ICT) ecosystems, connecting to external networks for control, monitoring, and maintenance. This

- connectivity introduces new vulnerabilities, particularly with the increasing use of robot APIs and the management of robots via portable devices like smartphones;
- Human–robot interaction. The shift towards software-based safety mechanisms over hardware solutions increases vulnerability to security incidents [15]. Next-generation industrial robots designed to work closely with humans further expand the scope of security attacks, posing direct threats to human safety;
 - Inadequate security measures. Surveys indicate a significant portion of robotic systems lack adequate security measures [1,16]. Many operators modify default safety settings, fail to implement access controls, and do not conduct regular security assessments, leading to increased risk.

2.2. State of the Art in Security Assessment

Ensuring security in robotic systems requires a comprehensive approach, incorporating threat modeling and robust security assessment methodologies.

2.2.1. Threat Modeling

Threat modeling is crucial for identifying and mitigating potential threats early in the system design process [17,18]. It involves examining the system from an adversary's perspective to determine what needs protection and from whom. Key steps include system description, architecture dataflow, the identification of trust boundaries, threat analysis, and the definition of countermeasures. Various threat modeling methods, such as STRIDE, PASTA, LINDDUN, and CVSS, provide different perspectives and approaches to assessing and addressing security risks. Tools like Cairis, Microsoft Threat Modeling Tool, OWASP Threat Dragon, Threagile, and Tutamantic facilitate these processes.

2.2.2. Security Knowledge Repositories

Several repositories provide valuable information for security assessment, including the following:

- CVE (Common Vulnerabilities and Exposures) [19]—Offers identifiers for computer security flaws, facilitating easy recognition and communication of vulnerabilities;
- NVD (National Vulnerability Database) [20]—Supplements CVE by providing additional information such as severity scores, countermeasures, and affected software configurations;
- CWE (Common Weakness Enumeration) [21]—Lists weaknesses in software and hardware, providing detailed descriptions and relationships with other weaknesses;
- CAPEC (Common Attack Pattern Enumeration and Classification) [22]—Classifies known attack patterns, aiding the understanding of how system weaknesses can be exploited;
- RVD (Robot Vulnerability Database) [23]—Focuses on vulnerabilities and bugs specific to robots' software and hardware. It uses the Robot Vulnerability Scoring System (RVSS) [23] to rate vulnerabilities, assisting in the prioritization and management of robot security concerns.

2.3. Related Works in Security Assessment in Robotic Systems

The security assessment of robotic systems is a multifaceted challenge requiring a combination of threat modeling, comprehensive security assessment methodologies, and the use of detailed security knowledge repositories. As robotic systems continue to evolve and integrate more deeply into various sectors, ensuring their security becomes increasingly critical to prevent potential cyber-attacks and safeguard human safety. Robust security

measures and continuous monitoring are essential to protect these systems from emerging threats and vulnerabilities.

The integration of sensors, actuators, interfaces, and information processing in robots introduces new vulnerabilities. Several studies have explored the security of robotic systems, as follows [16,17]:

- Cyber-physical honeypots—One study employed a cyber-physical honeypot using ROS to discover vulnerabilities and means of exploitation;
- Automobile hacking—Another study demonstrated hacking a modern automobile, compromising its digital dash, door locks, brakes, and engine control components;
- UAV attacks—Research on unmanned aerial vehicles (UAVs) revealed the impact of denial-of-service attacks on UAV cameras and network latency;
- Multi-robot systems—A model representing the performance of multi-robot systems highlighted how denial-of-service attacks could compromise cloud-robotic platforms;
- Specific robot assessments—Security assessments of specific robots, such as Pepper and Franka Emika Panda, uncovered vulnerabilities that could enable credential spoofing, data theft, and the hacking of connected devices.

These studies emphasize the need for robust security measures to address vulnerabilities arising from the integration of various technologies in robotic systems.

The FISHY approach [24] provides an automated cybersecurity threat remediation framework that integrates Cyber Threat Intelligence (CTI) to enhance security response mechanisms. The methodology revolves around Remediation Recipes, which define sequences of security actions for mitigating network-based threats. These recipes follow a structured meta-model, ensuring they are both human-readable and machine-actionable. Once identified, threats are processed by an Interpreter, which refines abstract recipes into CACAO Security Playbooks, a standardized format for automatic enforcement. A Deployment Engine then translates playbooks into enforceable security policies, modifying the network landscape as needed. The approach supports threat-sharing through the MISP platform, enabling organizations to exchange remediation strategies and enhance collective cybersecurity resilience. By integrating automated risk assessment, policy enforcement, and collaborative threat-sharing, FISHY strengthens network defenses, reducing reliance on manual intervention while improving response efficiency.

The MITIGATE approach [25] introduces a dynamic and collaborative risk assessment methodology tailored for cyber threats in supply chain ecosystems, particularly in maritime environments. It builds upon existing cybersecurity standards like ISO 28001, ISO 27005, and ISO 31000 while addressing interdependencies between business partners and their interconnected ICT assets. The methodology employs a graph-based risk modeling approach, where cyber assets and their interconnections are represented as a directed graph. Attack paths are analyzed using propagation rules, defining how vulnerabilities can be exploited and how threats can cascade through the supply chain. MITIGATE assesses risks at multiple levels, including individual, cumulative, and propagated vulnerabilities, impacts, and risks, ensuring a holistic understanding of potential attack scenarios. It integrates real-time threat intelligence from repositories like NVD and CERT databases and estimates zero-day vulnerabilities using machine learning techniques. Additionally, a game-theoretic approach is applied to determine optimal mitigation strategies, balancing defensive actions against potential adversarial attacks. By enabling collaborative risk assessment, MITIGATE allows business partners to share security insights and develop collective countermeasures, improving overall supply chain resilience.

The Eisenhower Matrix approach [26] in cybersecurity assessment provides a structured method for prioritizing security tasks in industrial environments, particularly in Industry 4.0 and smart manufacturing systems. This methodology categorizes security

actions based on urgency and importance, ensuring efficient resource allocation for threat mitigation. The matrix is divided into four quadrants, as follows: (i) Do—immediate actions such as software updates, IDS signature management, and security awareness programs; (ii) Plan—strategic activities like Red Team exercises, vulnerability management, and security education; (iii) Delegate—tasks requiring external oversight, including compliance updates and third-party risk management; and (iv) Eliminate—redundant policies and outdated security practices. By integrating the Zero Trust security model, the framework emphasizes continuous verification and adaptive risk management, reducing the attack surface in cyber–physical systems (CPS) and collaborative robotics. The approach enhances real-time security monitoring, incident response, and proactive risk mitigation, ensuring that cybersecurity efforts align with operational priorities in industrial settings.

The RFBR approach [27] presents a structured methodology for assessing information security risks in robotic systems, addressing their distinct characteristics compared to traditional industrial control systems (ICS) and IT networks. The framework introduces a multi-layered risk assessment model, focusing on initial security evaluation, attack feasibility analysis, and threat impact assessment. The methodology begins with an assessment of the structural and functional characteristics of the robotic system, considering factors such as hardware performance, software vulnerabilities, communication channels, and navigation integrity. A key innovation is the introduction of a three-tier criticality classification (high, medium, low), which quantifies the severity of security threats based on potential system disruption and impacts on national security, economy, and reputation. The framework also incorporates real-world cyber incidents, such as UAV hijacking and robotic system takeovers, to validate risk modeling. Additionally, the RFBR methodology integrates threat modeling with adversarial capability assessment, considering multi-stage cyber-attacks and their cascading effects. The approach emphasizes machine learning-driven automation to enhance real-time threat detection and mitigation strategies. By bridging risk analysis, security assessment, and proactive countermeasures, RFBR provides a comprehensive cybersecurity framework tailored for autonomous and interconnected robotic systems.

The paper presents the SESAME approach. It introduces a structured security assessment methodology designed for multi-robot systems (MRS), integrating automated threat detection, vulnerability assessment, and real-time mitigation strategies. The methodology leverages structured security repositories like CVE, CWE, and CAPEC to identify vulnerabilities and map them to potential attack patterns. A key component is the generation of Template Attack Trees, which model potential cyber–physical attack scenarios and assist in developing countermeasures. The Executable Digital Dependability Identity (EDDI) framework extends traditional risk assessment by incorporating real-time monitoring, anomaly detection, and automated response mechanisms using Intrusion Detection Systems (IDS) and MQTT-based communication protocols. SESAME employs tools like OpenVAS for vulnerability scanning, and integrates threat intelligence feeds to ensure dynamic risk assessment. The methodology has been successfully tested in critical infrastructures such as power stations and healthcare environments, demonstrating its effectiveness in securing interconnected robotic systems against evolving cyber threats. By combining design-time and runtime security evaluation, SESAME enhances the resilience, adaptability, and dependability of autonomous robotic systems in high-risk operational settings.

Table 1 summarizes the main features of these solutions. Each approach offers unique advantages based on its target domain and security assessment focus. SESAME excels in multi-robot security, leveraging real-time attack detection and automated responses. FISHY emphasizes policy-driven threat remediation and intelligence sharing, making it ideal for network-related assessments. MITIGATE is well-suited for supply chain security, using

graph-based modeling to assess cascading cyber risks. The Eisenhower Matrix focuses on cybersecurity task prioritization, ensuring optimal resource allocation in industrial settings. Lastly, RFBR is tailored for autonomous robotic security, integrating AI-driven risk assessment and multi-stage attack modeling.

Table 1. Comparative analysis for security assessment methodologies for robotic systems.

Feature	SESAME	FISHY	MITIGATE	Eisenhower Matrix	RFBR
Scope	Multi-Robot Systems (MRS)	Network Security and Threat Remediation	Supply Chain Risk Assessment	Industry 4.0 Cybersecurity	Autonomous Robotic Systems
Threat modeling	Template Attack Trees	Cyber Threat Intelligence (CTI)	Graph-Based Risk Modeling	Risk Prioritization	Initial Security Evaluation
Risk assessment	Automated Vulnerability Scanning	Threat Intelligence and Remediation Recipes	Attack Propagation and Game Theory	Security Task Prioritization	Multi-Stage Attack Feasibility
Real-time monitoring	EDDI-Based Security Management	MISP Threat Sharing and Policy Enforcement	Zero-Day Vulnerability Prediction	Integrated Zero Trust Approach	AI- and Machine Learning-Based Threat Analysis
Mitigation strategy	Automated Response via Security EDDIs	CACAO Security Playbooks	Collaborative Risk Assessment	Resource Allocation for Critical Security Tasks	Dynamic Risk Classification
Application domain	Power Stations, Healthcare Robots	Large-Scale Networks	Maritime and Logistics	Smart Manufacturing Systems	Autonomous Mobile Robots (AMRs)

SESAME introduces a novel, automated security assessment framework tailored for MRS, integrating real-time monitoring, automated vulnerability detection, and dynamic risk mitigation. Unlike other methodologies, SESAME uniquely leverages EDDIs to provide continuous security monitoring and automated responses based on detected threats. Its Template Attack Trees enhance threat modeling, offering structured, scenario-based risk assessments that surpass static vulnerability detection approaches like those in MITIGATE and RFBR. Unlike FISHY, which focuses on network-based security automation, SESAME integrates cyber–physical security considerations, making it ideal for autonomous robotic systems in critical infrastructures. Additionally, it outperforms the Eisenhower Matrix by automating risk prioritization, eliminating reliance on manual security task categorization. By combining automated security analysis, real-time threat detection, and adaptive response mechanisms, SESAME offers a comprehensive, scalable, and proactive security solution that is better suited for the evolving cybersecurity challenges in robotic and CPS.

3. The SESAME Security Methodology

This section describes the developed security assessment methodology for robotic systems that has been defined under the SESAME project.

3.1. Overview

The threat modeling process plays a crucial role in defining the security design and selecting appropriate security technologies for a system, considering its specific security requirements. The security assessment conducted within the context of SESAME is heavily influenced by the threat modeling process and adopts its fundamental principles. The SESAME security assessment follows a well-structured set of steps that often overlap with the systematic process of threat modeling, which has clearly defined steps based on the chosen model. The methodology diagram in Figure 1 illustrates these steps, including their inputs, outputs, external resources, and processes.

While the high-level steps outlined in the following sections provide a general approach, the key to effectively applying the proposed methodology to individual MRSs with unique requirements lies in providing a detailed description of the specific system under consideration. Factors such as the importance of assets and the delineation of trust boundaries containing these assets help to capture the distinct security requirements of

each system. Additionally, the identification of vulnerabilities and their combinations contributes to each system being a unique use case, resulting in varying outputs from the SESAME security methodology, such as potential attack scenarios and corresponding mitigations.

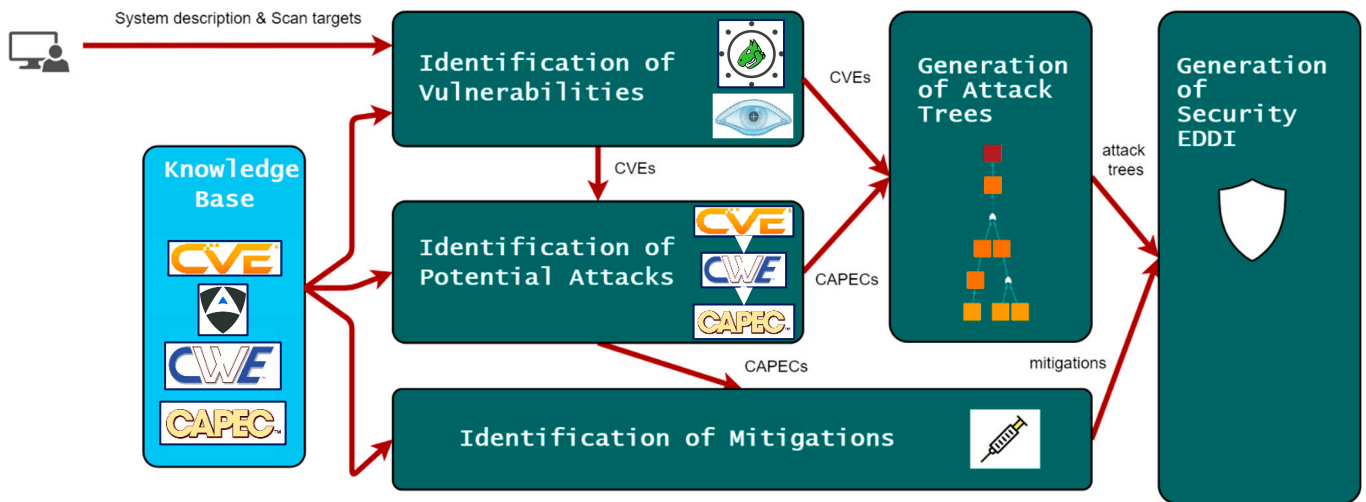


Figure 1. SESAME security methodology.

The SESAME security methodology is a comprehensive framework designed to address the unique security challenges faced by multi-robot systems (MRS). It emphasizes a structured approach to security assessment, leveraging state-of-the-art tools, techniques, and repositories to systematically identify and mitigate potential threats. The methodology integrates elements of threat modeling, vulnerability assessment, and attack simulation to provide robust security solutions tailored to specific system requirements.

3.2. Stage 1—System Description

The initial phase involves gathering detailed information about the target system. This includes understanding the system's purpose, components, architecture, and scope. Key questions are addressed to identify critical functions, potential vulnerabilities, and compliance requirements. The collected data are organized into categories such as Purpose, Components, Architecture, and Scope to facilitate a thorough understanding of the system's security landscape.

3.3. Stage 2—Identification of Vulnerabilities

This process utilizes publicly available repositories like CVE and RVD to identify known vulnerabilities in system components. Automated tools such as vulnerability scanners (e.g., OpenVAS, OPENSCAP) are employed to discover exposed services and known vulnerabilities. The output is a list of vulnerability identifiers relevant to the target system, which is continuously updated to include the latest information.

3.4. Stage 3—Identification of Potential Attacks

The vulnerabilities identified are mapped to potential attacks using additional resources like the CWE and CAPEC catalogs. This mapping process helps trace specific attack patterns (CAPEC-IDs) from identified vulnerabilities (CVE-IDs), providing a clear path from vulnerabilities to potential exploitation scenarios [28].

3.5. Stage 4—Identification of Mitigations

For each identified attack pattern, mitigation actions are determined. These actions are derived from the CAPEC repository, which includes specific protection mechanisms for documented attacks. This process ensures that appropriate countermeasures are identified for each potential attack scenario, enhancing the system's overall security posture.

3.6. Stage 5—Construct Template Attack Trees

Template Attack Trees are predefined attack patterns that describe common methods used by attackers. These templates are tailored to each system's unique characteristics, providing relevant attack scenarios based on identified vulnerabilities. The attack trees help visualize the steps an attacker might take to compromise the system, and are used to model potential attack scenarios and develop corresponding mitigation strategies.

3.7. Stage 6—Generation of Attack Trees

Attack trees are generated by leveraging relationships between different attack patterns in the CAPEC repository (e.g., as shown in Figure 2). These relationships, such as "CanFollow" and "CanPrecede", help construct sequential attack scenarios. Template Attack Trees are also utilized to merge different graphs, providing a comprehensive view of potential attack paths and their respective mitigations.

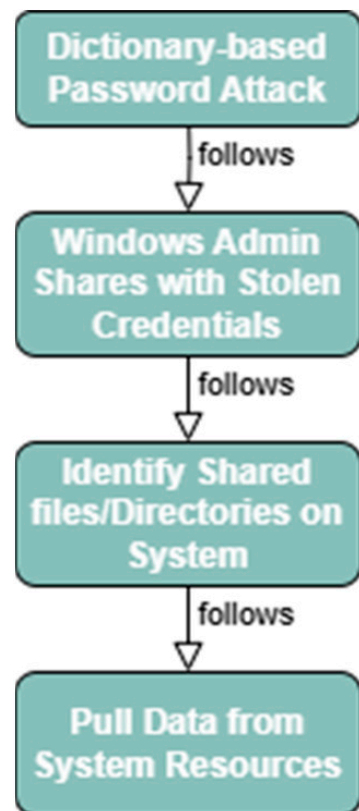


Figure 2. Example graph that can be produced utilizing the CanFollow relationship of CAPEC.

3.8. Stage 7—Generation of Security EDDIs

The final step involves creating Executable Digital Dependability Identities (EDDIs). EDDIs extend the concept of Digital Dependability Identities (DDIs) by incorporating real-time monitoring and management capabilities. They facilitate online monitoring, runtime diagnostics, dynamic risk prediction, and recovery planning [29]. EDDIs also enable communication with other EDDIs to ensure coordinated dependability management across the MRS.

3.9. Safety and Security Integration

The SESAME methodology recognizes the interplay between safety and security. It ensures that security measures protect against unauthorized access and malicious manipulation, while safety mechanisms safeguard human well-being and prevent potential harm. EDDIs integrate security and safety information to provide a holistic view of the system's dependability, making recommendations to mitigate risks based on identified threats.

The SESAME security methodology provides a structured and systematic approach to securing multi-robot systems. Integrating threat modeling, vulnerability assessment, attack simulation and real-time monitoring ensures robust protection against a wide range of security threats. The methodology's comprehensive framework helps achieve the overall dependability goals of robotic systems, enhancing their resilience against both cyber and physical attacks.

4. Implementation

The SESAME security methodology is designed to systematically address the security challenges faced by MRS. This methodology integrates various tools and techniques to gather security information, identify vulnerabilities, and assess potential attacks. The ultimate goal is to create EDDIs that incorporate both security and safety information for runtime usage. This detailed implementation summary outlines the steps and tools used in this process, providing a comprehensive view of the SESAME security assessment framework.

4.1. System Description

The first step in the SESAME security methodology is to gather detailed information about the target system. This involves using two primary methods—a user interface and OpenVAS v22, an automated scanner.

4.1.1. User Interface (UI)

The UI is a web-based application developed in Java with Bootstrap, a popular HTML, CSS, and JavaScript framework for creating responsive, mobile-first websites. The UI consists of forms and questionnaires designed to capture specific details about the system architecture, components, assets, entry points, and trust boundaries (Figure 3). By guiding system administrators through a step-by-step process, the UI ensures that all necessary information is gathered consistently and in a standardized format. This structured approach helps streamline the subsequent steps in the security assessment methodology, enhancing the efficiency and effectiveness of the process.

Defining the communication protocols that are used among the system components allows the SESAME security assessment to reveal vulnerabilities that are not present in any individual component. This is especially relevant in the context of multi-robot systems where communication plays a vital role. The communication among the robots allows them to create swarms and work together, and at the same time, it introduces new vulnerabilities and new attack surfaces. Questions that contribute towards an understanding of system architecture include the following:

1. What is the overall design of the system?
2. How are the system components connected?
3. Which are the system access points?
4. What is the path that data follow? What is the input and the output of the system?
5. Are there any third-party integrations into the system?
6. Is the system monitored?

System description - Wizard ✕

System Purpose
Step-1

System Components
Step-2

System Architecture
Step-3

System Scope
Step-4

What is the primary function of the system? Does it perform a single task or multiple tasks?

Is the system critical to the organization's operations?

Does the system serve a specific business goal?

What would be the impact of system unavailability?

Are there any compliance requirements associated with the system?

Who are the users of the system and what roles do they have?

Are there any known system vulnerabilities?

Which parts of the system are considered the most critical?

Figure 3. Questionnaire wizard.

Defining the desired scope of the security assessment determines the extent of the analysis. The SESAME security assessment can be conducted on various levels, ranging from the entire target system to specific subsystems or individual components, depending on specific security requirements. Factors that may influence the security assessment include the system's complexity, the level of potential risk, the available resources, and the specific requirements or regulations governing the system. Questions that fit this category include the following:

1. What are the boundaries of the system to be assessed?
2. What is the acceptable security level for the system to be assessed?
3. Are there any compliance requirements for the system to be assessed?

4.1.2. OpenVAS

Open Vulnerability Assessment Scanner (OpenVAS) [30] is an automated scanner that identifies active services and their vulnerabilities within a network. OpenVAS meticulously

scans all ports on the target system, providing comprehensive reports on discovered assets, such as running software and specific version numbers. It also conducts attacks using a plethora of known exploits, reporting on the vulnerable services with high-level descriptions, CVE scores, and severity levels. OpenVAS supports various scanning configurations, including fast, deep, and custom scans, enhancing its coverage and effectiveness.

The installation of OpenVAS is straightforward, involving commands to clone the code from GitHub v2 and create a Docker container to run the scanner. Once installed, OpenVAS operates several components, including the scanner, manager, and a web interface, providing a robust platform for vulnerability assessment.

4.2. Identification of Vulnerabilities

The SESAME methodology utilizes two main repositories to identify known vulnerabilities in system components—CVE and RVD. Parsers for these repositories search for vulnerabilities based on the identified components' names and versions.

4.2.1. CVE-Search for Vulnerabilities Identification

CVE-search is an open-source tool that imports CVE and CPE data into a local MongoDB instance, facilitating fast and secure searches. It allows users to search for specific vulnerabilities, explore detailed information about them, and track their status and associated resources. CVE-search offers various query options, such as searching for vulnerabilities by product name, specific CVE IDs, or text searches in vulnerability summaries. This tool is widely used by organizations, including CERTs and CSIRTs.

CVE-search is used by many organizations including the public CVE services of Computer Incident Response Center Luxembourg (CIRCL). The source code is available on GitHub. A whole community maintains, it including CIRCL.

There are different ways to form a request asking for vulnerabilities, as follows:

- Request returning vulnerabilities directly assigned to a specific product (`./bin/search.py -p microsoft:windows_7 -a -o json`);
- Request returning vulnerabilities based on text search in the vulnerability summary (`./bin/search.py -f "robotic simulator" -a -o json`);
- Request for a specific CVE ID (`./bin/search.py -c CVE-2010-3333`);
- Request the last two CVE entries in atom format (`./bin/dump_last.py -f atom -l 2`).

4.2.2. RVD Custom Parser

The RVD custom parser extends the capabilities of the RVD project by providing functionalities tailored to SESAME's needs. The RVD parser can search for vulnerabilities based on product descriptions or CPE identifiers and query related CWEs for specific CVEs. This parser ensures that the unique complexities and characteristics of robots are adequately addressed, enhancing the overall security assessment.

A REST API is created for the RVD parser to update the local version of the RVD database. This API allows for regular updates to include newly added vulnerabilities, ensuring the vulnerability database remains current.

To achieve the desired functionality described in the previous paragraph, a custom RVD parser has been created. The RVD installation offers the `"rvd list --dump --label vulnerability"` command that returns all the RVD database entries, which are labeled as vulnerabilities. An example of such an entry is depicted in Listing 1 below. The information provided for each robot vulnerability includes related CVEs and CWEs, affected systems, severity scores (RVSS, CVSS), exploitation and mitigation descriptions.

Listing 1. Example robot vulnerability from the RVD database.

```

id: 3337
title: Service DoS through arbitrary pointer dereferencing on KUKA
simulator
type: vulnerability
description: "Visual Components (owned by KUKA) is a robotic
simulator that allows simulating factories and robots in order
to improve planning and decision-making processes. ... Accordingly, a
DoS in the simulation might have higher repercussions, depending on the
Industrial Control System (ICS) ICS infrastructure."
cwe: CWE-248
cve: CVE-2020-10292
keywords:
- KUKA, RMS sentinel LM, Visual Components, DoS
system: Visual Components Network License Server 2.0.8
vendor: KUKA Roboter GmbH, Visual Components
severity:
  rvss-score: 6.1
  rvss-vector: RVSS:1.0/AV:IN/AC:L/PR:N/UI:N/S:U/Y:Z/C:N/I:L/A:H/H:N
  severity-description: High
  cvss-score: 8.2
  cvss-vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:H
links:
- https://cwe.mitre.org/data/definitions/248.html
- https://www.visualcomponents.com/products/downloads/
- https://www.visualcomponents.com/products/visual-components/
flaw:
  phase: runtime-operation
  specificity: subject-specific
  architectural-location: application-specific
  application: Visual Components, RMS sentinel LM
  subsystem: simulation
  package: null
  languages: null
  date-detected: null
  detected-by: Sharon Brizinov (Claroty)
  detected-by-method: testing-dynamic
  date-reported: 2020-10-27
  reported-by: Sharon Brizinov (Claroty)
  reported-by-relationship: security researcher
  issue: https://gitlab.com/aliasrobotics/offensive/rvd/flaws/-
/issues/712
  reproducibility: always
  trace: null
  reproduction: null
  reproduction-image: null
exploitation:
  description: |
    To exploit this vulnerability the attacker needs to have network
    access to the license server (either because it's exposed or because
    the internal network has been compromised. Cause is related to the
    number of requested strings to merge, which is not correlated to the
    number of strings provided, and so arbitrary pointers from the stack
    are popped out and dereferenced. This results with an uncaught Access
    Violation exception which terminates the program. PoC available
    constructs a response reply to featureInfoToFile with is a mismatch
    between the number of strings to merge and the requested amount
    leading to an Access Violation exception and terminating the program.
    See alurity's robotsplit/exploits/kuka/rms exploits.
  exploitation-image: Not available
  exploitation-vector: null
  exploitation-recipe:
    networks:
    - network:
      - driver: bridge
      - name: kuka-simulation
      - subnet: 14.0.0.0/24
    vms:
    - vm:
      - name: vml
      - path: $(pwd)/vms/visualcomponents_2.0.8
      - network: kuka-simulation
      - ip: 14.0.0.4
    containers:
    - container:
      - name: attacker

```

```

- modules:
  - base:
registry.gitlab.com/aliasrobotics/offensive/alurity/alurity:latest
  - volume:
registry.gitlab.com/aliasrobotics/offensive/alurity/expl_robosploit/e
xpl_robosploit:latest
  - volume:
registry.gitlab.com/aliasrobotics/offensive/alurity/deve_atom:latest
  - volume:
registry.gitlab.com/aliasrobotics/offensive/alurity/reco_nmap:latest
  - volume:
registry.gitlab.com/aliasrobotics/offensive/alurity/expl_icssploit:la
test
  - volume:
registry.gitlab.com/aliasrobotics/offensive/alurity/expl_metasploit:l
atest
  - volume:
registry.gitlab.com/aliasrobotics/offensive/alurity/fore_wireshark:la
test
  - network: kuka-simulation
mitigation:
  description: |
    Do not launch Visual Components while connected to local or wide
    area networks. Contain the simulation through
    virtualization.
  pull-request: null
  date-mitigation: null

```

The whole set of available robot vulnerabilities is used as the input for our custom parser. A set of Java classes has been created for storing and managing the information provided for the incoming robot vulnerabilities (Figure 4). The main class is called “Rvd-Vulnerability”, while four more subclasses are needed, called “Severity”, “Exploitation”, “Flaw”, and “Mitigation”.

4.3. Identification of Potential Attacks

The vulnerabilities identified in the previous step serve as input for determining potential attacks using the CVE-search and a CAPEC custom identifier.

4.3.1. CVE-Search for Attack Identification

CVE-search also helps identify related attacks for given vulnerabilities by providing lists of CAPEC IDs in its output. This tool can be queried for known attacks related to a specific CVE-ID or product, offering a comprehensive view of potential threats.

4.3.2. CAPEC Custom Identifier

The CAPEC custom identifier uses a local instance of the CAPEC catalogue to identify known attacks based on provided CWEs. A REST API allows for querying the local CAPEC database instance, returning relevant attack patterns. This tool ensures that the identified attacks are closely aligned with the specific weaknesses of the target system, providing targeted mitigation strategies. The structure of the CAPEC class can be seen in Figure 5.

4.4. Generation of Attack Trees

Attack trees are generated in two steps—creating small trees based on CAPEC relationships and utilizing Template Attack Trees.

4.4.1. Small Attack Trees

Small attack trees are based on “CanFollow” and “CanPrecede” relationships between CAPEC attack patterns [28,29]. A Java class stores the information, and recursive methods parse the trees to identify potential attack scenarios. These relationships help construct sequential attack scenarios, providing a detailed view of how an attacker might exploit identified vulnerabilities.

4.4.2. Template Attack Trees

Template Attack Trees are predefined attack patterns that incorporate both cyber and physical vulnerabilities. After identifying potential attacks, the list of attacks is compared with template tree leaves to identify relevant attack scenarios. These trees visualize attack paths, showing the steps an attacker might take to compromise the system. They help security experts develop effective mitigation strategies by revealing common attack methods and techniques.

The implementation of these trees allows for recursive parsing, with methods that check each node and its children against the identified potential attacks. This process ensures that the generated attack trees are comprehensive and relevant to the target system. Figure 6 illustrates a Template Attack Tree for a robotic system with cyber and physical vulnerabilities.



Figure 4. RVD Java classes of the custom RVD parser.

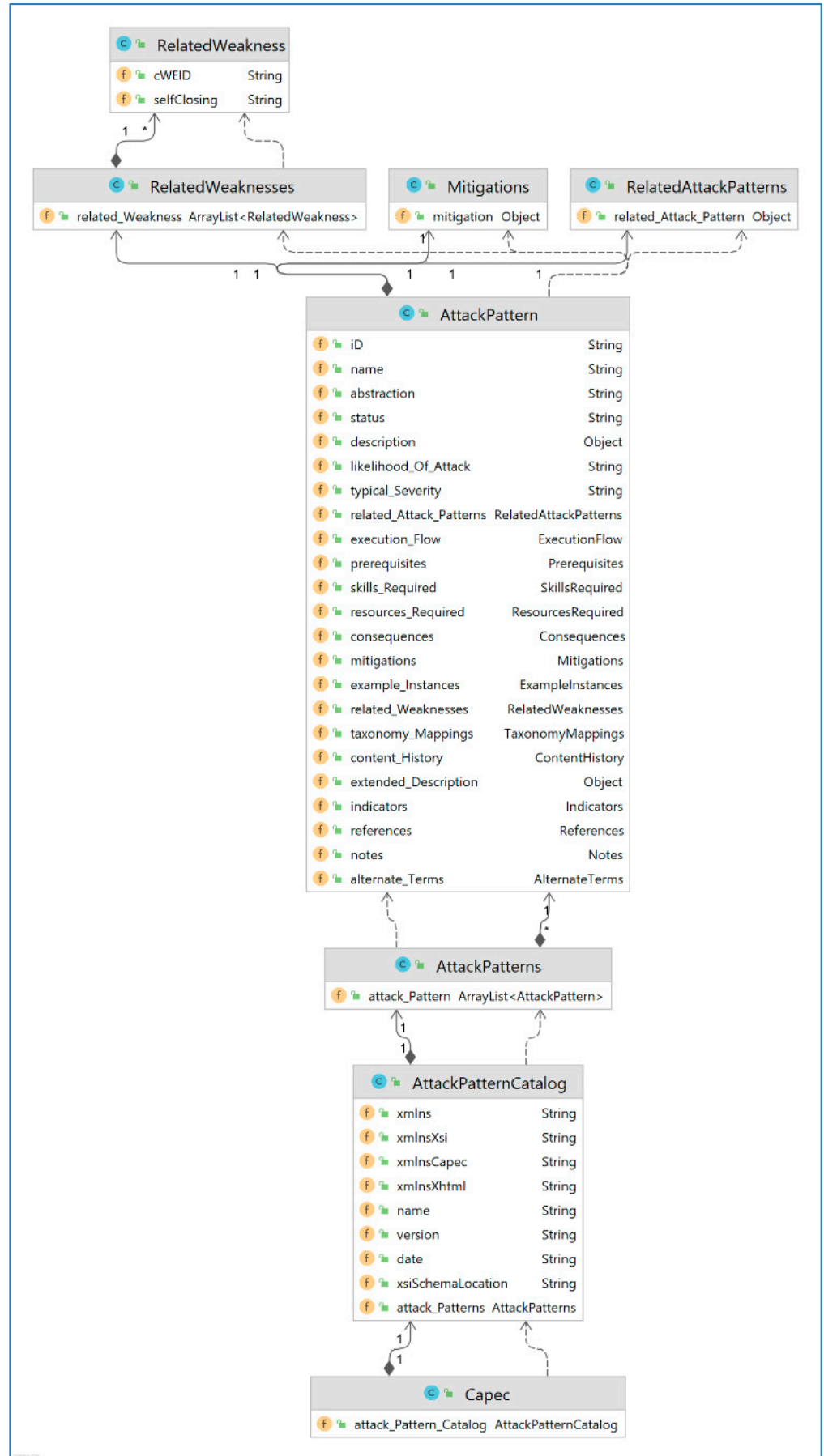


Figure 5. CAPEC classes of the custom CAPEC identifier.

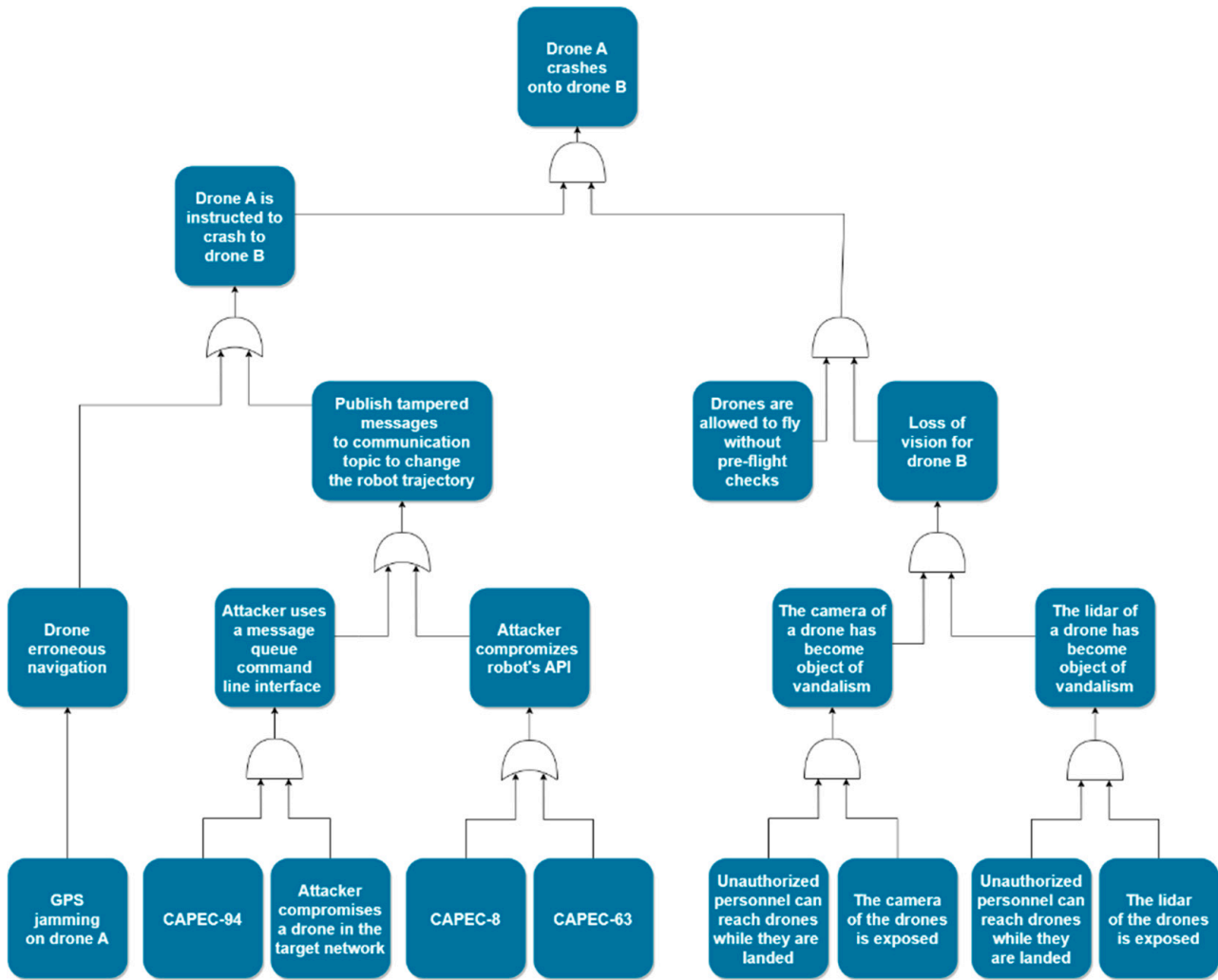


Figure 6. Template Attack Tree with cyber and physical vulnerabilities.

4.5. Generation of Security EDDIs

The identified potential attack trees are translated into security EDDIs, which consist of Python v3.13 scripts corresponding to each attack tree. These scripts interact with an MQTT broker and an Intrusion Detection System (IDS) to monitor and respond to threats.

4.5.1. Intrusion Detection System (IDS)

An IDS is a monitoring system designed to detect suspicious activities in the communication between a system and its external entities. SESAME uses Snort v22 [31], a signature-based IDS, to monitor incoming network traffic for malicious packets. Snort generates alerts for detected suspicious activities, which are then published to MQTT topics. The Python scripts listen to these alerts and parse the attack trees to identify ongoing attacks and their potential impacts.

Snort employs a rule-based approach to define malicious network activity. It triggers alerts when a rule is matched, logging the events and allowing for detailed analysis. Snort’s flexibility and wide adoption make it an ideal choice for SESAME’s security monitoring needs.

4.5.2. Python Scripts

The Python scripts hold the logic for identifying the ultimate goal of an attacker based on the information from the attack trees and the alerts generated by the IDS. Each script

listens to the MQTT topic for IDS alerts and parses the attack tree based on the parent/child relationships of the attacks. If the script detects a match with the identified potential attacks, it continues parsing up the tree to determine if the attacker's ultimate goal can be achieved.

The distributed nature of the EDDI solution allows for the easy integration of additional sensors or monitoring tools. For example, sensors that detect physical attacks can be incorporated by publishing alerts to an MQTT topic, which the corresponding Python scripts can then monitor.

4.6. Overall Offerings

The SESAME security methodology implementation integrates various tools and techniques to provide a comprehensive security assessment for multi-robot systems. By leveraging automated scanners, vulnerability repositories, and attack trees, SESAME ensures robust protection against potential threats. The creation of security EDDIs facilitates real-time monitoring and management, enhancing the overall dependability of robotic systems. This structured and systematic approach ensures that both cyber and physical vulnerabilities are addressed, providing a holistic security solution for MRS.

5. Demonstration

The proposed methodology has been applied in two piloting use cases that are presented in the following subsections.

5.1. Power Station Inspection Using Autonomous MRS

The piloting scenario involves the inspection of a power station, in Cyprus, responsible for the country's electricity generation. This piloting scenario was chosen due to the presence of high-risk locations, including fuel tanks, turbines, boilers, and water pumps, which require regular inspection and monitoring to ensure safety and security. The scenario involves two operation modes—normal and emergency.

In normal mode, the power station undergoes routine inspections using unmanned aerial vehicles (UAVs) to detect anomalies and ensure smooth operations [32]. In emergency mode, such as after an earthquake, multiple UAVs are deployed to assess damage, locate injured individuals, and gather critical information while maintaining a safe distance from the disaster site. The emergency response involves establishing a coordination center and deploying UAVs to provide real-time situational awareness and support.

The SESAME project integrates multiple tools to enhance the security and reliability of UAV operations in both normal and emergency scenarios [27]. Figure 7 shows a view of the system using three UAVs.

5.2. Hospital Multi-Robot Intralogistics

PAL Robotics aims to enhance quality of life through service robotics and automation technologies [33]. The company designs and manufactures solutions for various industries, including healthcare, where autonomous mobile robots (AMRs) are used to support intralogistics tasks. This use case targets the deployment of an AMR team within a hospital environment, emphasizing safe and secure collaboration with humans. The goal is to showcase the effective integration of SESAME project developments in both simulated and real-world settings.

The scenario involves a fleet of TIAGo robots (PAL Robotics, Barcelona, Spain) operating within a hospital, performing healthcare-related tasks alongside humans. The fleet includes a TIAGo single-arm robot and TIAGo Base robots, managed by a fleet management application on an external PC. Figure 8 presents the available robot configurations. The robots share a common map and handle safety features individually. The deployment

involves testing safety and security features, including the ability to respond to human presence and potential cyber threats.

The development process includes a simulation phase using ROS Noetic LTS and Gazebo 11 in a Docker environment, followed by real-world testing in PAL Robotics' kitchen area. The simulation helps assess and integrate SESAME's capabilities, which are then validated in the real scenario to ensure consistency and effectiveness.

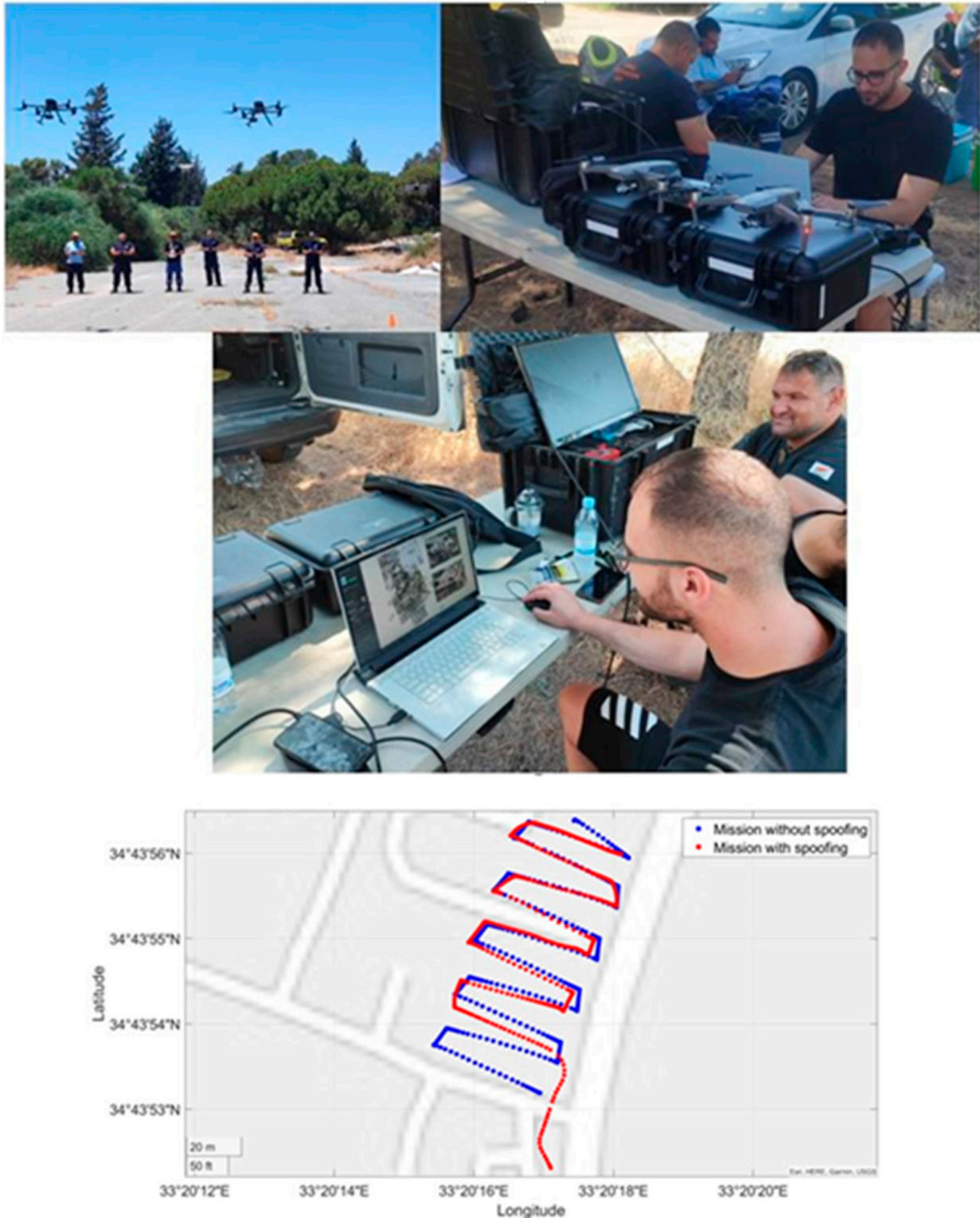


Figure 7. Area mapping with two UAVs (leader UAV and another UAV).

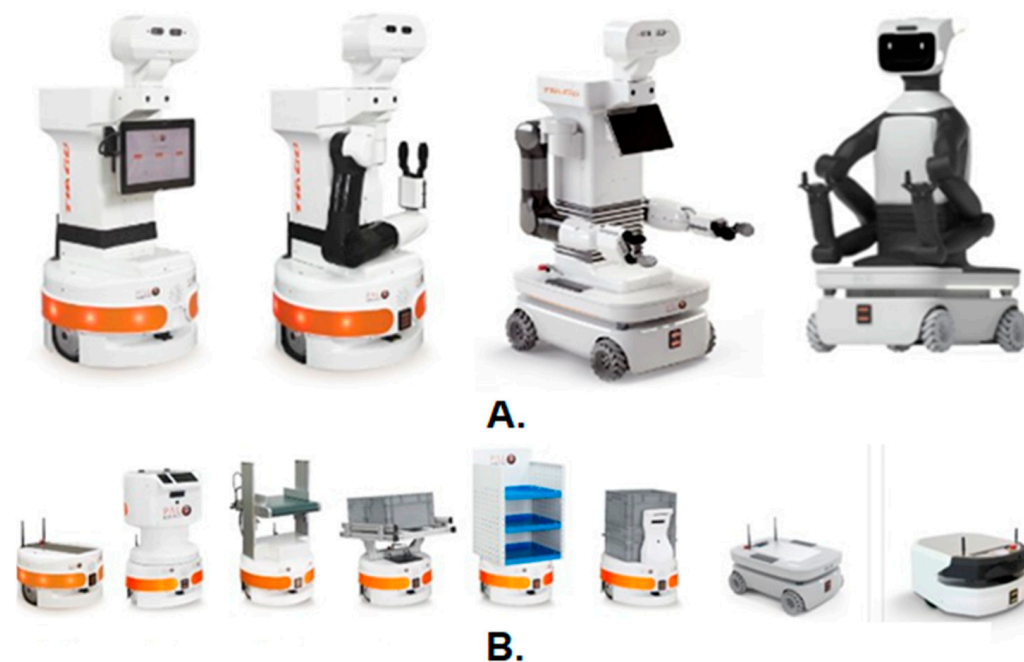


Figure 8. (A) TIAGo Family and (B) TIAGo BASE Family.

5.3. Evaluation

For security reasons, the specific vulnerabilities identified in the project's use cases cannot be disclosed. However, since all use cases involve the ROS, we focus on discussing publicly known ROS vulnerabilities, which are relevant across all scenarios. The SESAME security assessment methodology was applied to both use cases described above, as follows: (i) the inspection of a power station using UAVs under both normal and emergency conditions, and (ii) the secure operation of a fleet of AMRs with task exchangeability. Given this common technological foundation, a security assessment was conducted following SESAME's methodology, including vulnerability identification, attack mapping, and attack tree construction.

5.3.1. Identification of ROS Vulnerabilities

To systematically identify vulnerabilities in ROS, we utilized CVE-search, an open-source tool for querying publicly known vulnerabilities, and a custom RVD parser to extract additional security data. The analysis yielded 11 unique vulnerabilities affecting ROS, each categorized by its associated CVE identifier. The identified vulnerabilities are related to common weaknesses in various categories, including communication security issues, cryptographic flaws, memory safety vulnerabilities, and improper input validation. Notable examples include the following:

- CVE-2016-10681, which appears under two CWEs—CWE-300 (Channel Accessible by Non-Endpoint) and CWE-310 (Cryptographic Issues), indicating weaknesses in secure communication and cryptographic mechanisms;
- CVE-2019-13445 and CVE-2020-16124, both classified as CWE-190 (Integer Overflow or Wraparound), highlighting risks of improper arithmetic handling that could lead to memory corruption or privilege escalation;
- CVE-2019-13566, linked to CWE-120 (Buffer Copy without Checking Size of Input), representing a classic buffer overflow scenario;
- CVE-2019-19625 and CVE-2019-19627, both mapped to CWE-200 (Exposure of Sensitive Information to an Unauthorized Actor); exposing weaknesses in access control mechanisms.

- CVE-2020-10271, which falls under CWE-668 (Exposure of Resources to the Wrong Sphere), indicates potential unauthorized access to critical system components;
- CVE-2020-10272, classified as CWE-306 (Missing Authentication for Critical Function), highlighting a lack of necessary authentication mechanisms;
- CVE-2020-10289, mapped to CWE-20 (Improper Input Validation), which could lead to unexpected behavior due to unvalidated user inputs.

These vulnerabilities expose ROS-based systems to a range of attacks, necessitating an in-depth threat analysis.

5.3.2. Mapping Vulnerabilities to Attack Patterns

Following the identification of vulnerabilities, we mapped them into CAPEC attack patterns. This process enables the systematic analysis of how adversaries might exploit these weaknesses. The attack patterns associated with the identified CWEs reveal various security threats, including the following:

- Man-in-the-middle (MitM) attacks, linked to CWE-300, enable adversaries to intercept and manipulate communication channels. Related attack patterns include CAPEC-94 (Adversary in the Middle), CAPEC-466 (Bypassing Same-Origin Policy), and CAPEC-615 (Evil Twin Wi-Fi Attack), demonstrating potential risks in network communication;
- Integer overflow exploitation, associated with CWE-190, aligns with CAPEC-92 (Forced Integer Overflow), which adversaries could leverage to gain unauthorized control over memory regions;
- Buffer overflow vulnerabilities, mapped to CWE-120, correspond to CAPEC-10 (Buffer Overflow via Environment Variables), CAPEC-100 (Overflow Buffers), CAPEC-45 (Buffer Overflow via Symbolic Links), and CAPEC-67 (String Format Overflow in syslog), among others. These patterns emphasize how unsafe memory operations can be weaponized;
- Sensitive information exposure, associated with CWE-200, is linked to CAPEC-116 (Excavation), CAPEC-169 (Footprinting), CAPEC-224 (Fingerprinting), and CAPEC-497 (File Discovery), highlighting threats to confidentiality;
- Resource exposure vulnerabilities, mapped to CWE-668, indicate risks related to unauthorized access, but no specific CAPEC attack patterns were retrieved for this weakness;
- Authentication weaknesses, categorized under CWE-306, relate to attacks such as CAPEC-12 (Choosing Message Identifier), CAPEC-36 (Using Unpublished Interfaces), and CAPEC-62 (Cross-Site Request Forgery, CSRF), which can be exploited to bypass security controls;
- Improper input validation, associated with CWE-20, was found to correspond to a wide range of attack techniques, including CAPEC-88 (OS Command Injection), CAPEC-101 (Server-Side Include Injection), CAPEC-136 (LDAP Injection), and CAPEC-230 (XML Nested Payloads), among others.

The diversity of identified attack patterns highlights the complexity of potential cyber threats against ROS-based systems. Moreover, some attacks can serve as precursors to more sophisticated exploitation strategies, necessitating further analysis through attack tree modeling.

5.3.3. Attack Tree Construction

To better understand multi-stage attacks, we constructed attack trees using CanFollow and CanPrecede relationships between attack patterns. These relationships illustrate how certain attacks can enable subsequent exploitation steps. Four distinct attack trees were

formed, representing sequentially linked attack chains where the execution of one attack facilitates another.

What follows is the manual step of the construction of the Template Attack Trees. An example of such a tree can be seen in Figure 9. During this process, security experts create attack trees, which include a subset of the identified potential attacks. This can be done utilizing knowledge published in security-related literature or made available from real-life attack incidents of the past.

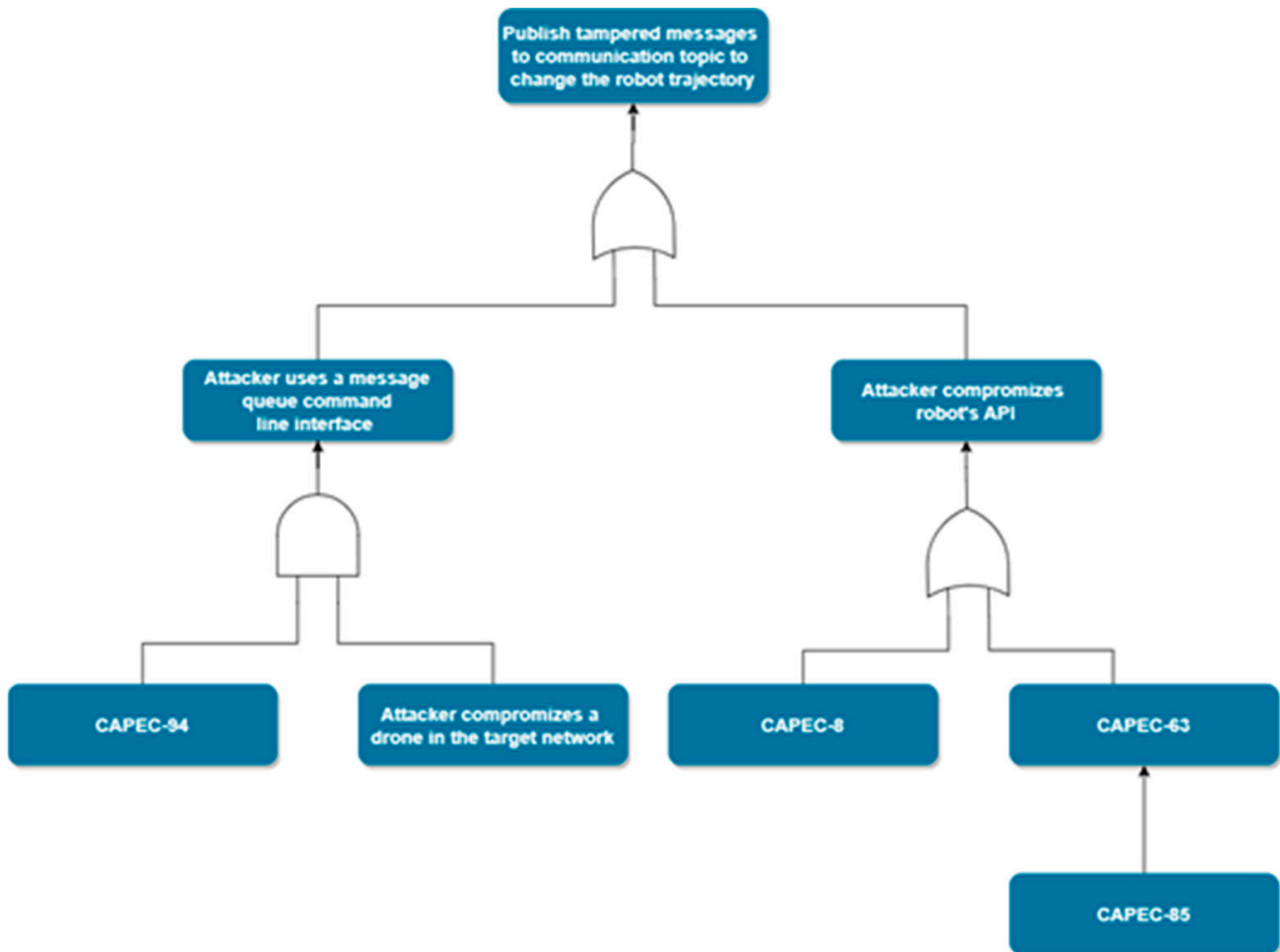


Figure 9. Example Template Attack Tree.

Figure 9 illustrates a Template Attack Tree, where the adversary's goal is to publish malicious messages to communication topics. The tree includes three known CAPEC attack patterns on its leaves.

The first attack, CAPEC-94 (Adversary in the Middle, AiTM), involves intercepting and modifying communication between system components. If combined with a compromised robot, which the attacker uses to publish messages, it enables message injection via a message queue command-line interface.

The second attack, CAPEC-8 (Buffer Overflow in an API Call), exploits vulnerable libraries used by software, rendering them susceptible to buffer overflows. The third, CAPEC-63 (Cross-Site Scripting, XSS), allows adversaries to embed malicious scripts, executing them with the victim's privileges. Either attack can lead to a compromised robot API.

Both security states—using a message queue CLI or compromising the robot API—allow the attacker to publish tampered messages, altering robot trajectories. Since message

queues facilitate trajectory commands in SESAME use cases, such manipulation could cause collisions, operational disruptions, or safety risks.

Finally, CAPEC-85 (AJAX Footprinting) acts as a precursor to CAPEC-63 (Cross-Site Scripting, XSS). This refinement underscores the cascading nature of security threats, where an initial reconnaissance attack (e.g., AJAX Footprinting) provides the necessary conditions for executing subsequent exploits (e.g., XSS).

5.3.4. Security EDDI Construction

To translate these attack trees into actionable security analysis, we developed a Python-based security EDDI script. The script monitors network events and intrusion alerts by subscribing to an Intrusion Detection System (IDS) alert topic (“snort”). It evaluates attack feasibility based on predefined conditions, including compromised system states and active attack indicators.

The script dynamically checks fault tree conditions and determines whether an attack scenario is unfolding. For instance, if CAPEC-94 (Adversary in the Middle) is active and a drone has been compromised, the system identifies a potential Message Queue CLI Interface Exploitation. Similarly, if either CAPEC-8 (Buffer Overflow in an API Call) or CAPEC-63 (Cross-Site Scripting) is detected, it triggers an alert indicating a compromised robot API, which could subsequently lead to the publication of tampered messages affecting robot trajectories.

This methodology bridges theoretical attack modeling with practical threat detection, enabling automated security assessment. While this example used only 11 vulnerabilities, the methodology scales effectively, uncovering a significantly larger attack surface in complex ROS-based environments.

5.3.5. Runtime Intrusion Detection

The security assessment involves deploying an IDS integrated with the security EDDI. Snort, an open-source IDS, is used to monitor network traffic for malicious activities. The system setup includes Snort, Barnyard (for data handling), MariaDB (for structured data storage), and a custom Python script (SnortPublisher) to facilitate communication between the IDS and the MQTT broker.

Snort examines network traffic based on predefined rules to detect suspicious patterns. Barnyard processes Snort’s binary logs, translating them into a structured format for storage in MariaDB. SnortPublisher v1.0 then extracts relevant data from the database and publishes them to an MQTT topic in JSON format. This setup enables the real-time monitoring and detection of security events, with the security EDDI analyzing alerts to identify and mitigate potential threats.

Two types of cyber-attacks were simulated to test the system’s resilience—spoofing ROS messages and denial-of-service (DoS) attacks.

In the Power Station Inspection use case, we used a spoofing attack by pinging a single drone while it was on a mission. Without integrating the security EDDI tool, after the attack, the trajectory of the affected UAV was shifted from the desired position. Figure 7 shows how the security attack can affect the mapping procedure by showing the deviation of the trajectory of the drone under attack (red color). The blue color indicates the correct trajectory of the drone with no attack.

On the other hand, when SESAME technologies were used, the attack was detected immediately by the security EDDI, and then ConSerts triggered Collaborative Localization for safe landing.

The use of the security EDDI improved the detection of specific security attacks such as the spoofing attack. Thus, the use of this tool provides 100% detection of such security attacks.

In the Hospital Multi-Robot Intralogistics use case, the spoofing attack was conducted using a laptop, which sent a ROS message to the 'pause_navigation' topic, causing the robot to stop its movement. The attack script repeatedly sent messages to disrupt the robot's navigation, demonstrating the system's vulnerability to unauthorized commands. Regarding the DoS Attack, a series of ROS messages with zero values was sent to the 'nav_vel' topic, causing the robot to temporarily stop each time a message was received. This prolonged the robot's journey to its destination, illustrating the impact of continuous message flooding on system performance.

One of the most notable improvements that security EDDI introduced was the reduction in attack detection time, particularly for denial-of-service (DoS) and spoofing attacks. By leveraging real-time monitoring and an intrusion detection system, the security EDDI allowed for a faster identification of security threats, ensuring that malicious activities were recognized and mitigated before they could disrupt operations. This capability greatly enhanced the system's responsiveness, reducing the risk of prolonged service interruptions caused by cyber threats.

Another improvement was the increase in system availability, allowing robots to maintain their operational capacity even in the presence of security incidents. The security EDDI enabled proactive threat response mechanisms that helped ensure the continuity of service, even when attacks were detected. This enhancement meant that the robotic fleet could operate with minimal downtime, reducing the likelihood of disruptions in the logistics workflow.

Additionally, the security EDDI contributed to the faster identification of potential issues in robotic missions, particularly those related to navigation and safety. By continuously analyzing system behavior and detecting anomalies early, it allowed for the early intervention and correction of problems before they could escalate into failures. This not only improved the efficiency of robotic operations, but also enhanced the overall security posture of the system.

The integration of SESAME's security assessment methodology significantly enhances the security and reliability of MRS operations. The key benefits include the following:

- Improved vulnerability management—The IDS setup and OpenVAS provide a systematic approach to identifying and mitigating security vulnerabilities, reducing the risk of exploitation by adversaries;
- Enhanced real-time security monitoring—The security EDDI ensures continuous monitoring and dynamic threat assessment, enabling proactive responses to security incidents;
- Increased system availability and reliability—The combined use of these tools helps maintain a high availability and reliability of MRS operations, crucial for both routine conditions and emergency responses;
- Comprehensive threat mitigation—By addressing both cyber and physical security threats, the SESAME framework ensures a holistic approach to safeguarding the power station and its critical infrastructure.

Overall, the application of the SESAME security assessment methodology in piloting environments with drones and healthcare robots demonstrates a robust and effective approach to managing security risks in complex and dynamic operational settings.

6. Scalability and Implementation Challenges Across Robotic Platforms

6.1. The Challenges

The SESAME security methodology is designed to be adaptable across a wide range of robotic platforms, from autonomous drones and industrial robotic arms to collaborative robots operating in dynamic environments. However, achieving scalability presents several challenges, particularly in accommodating the diverse architectures and operational constraints of different robotic systems [34].

One of the primary challenges lies in adapting the methodology to heterogeneous robotic architectures. Each platform has distinct hardware configurations, software frameworks, and communication protocols that influence its security vulnerabilities. To address this diversity, SESAME employs a modular security assessment approach that tailors threat models to the unique characteristics of each robotic system. The use of Template Attack Trees allows security analysts to generalize attack scenarios, making them applicable across different platforms with minimal modifications. Additionally, the introduction of Executable Digital Dependability Identities (EDDIs) ensures continuous security monitoring, adapting dynamically to the specific vulnerabilities of each robot.

Beyond architectural diversity, another critical challenge is the computational overhead associated with security assessments, particularly in real-time robotic applications where performance and responsiveness are paramount. Security evaluation processes, such as vulnerability scanning and attack detection, require computational resources that may strain embedded robotic systems with limited processing power. Real-time decision-making is crucial in robotics, where even minor delays in processing security checks could lead to operational inefficiencies or safety risks. To mitigate this, SESAME integrates lightweight intrusion detection mechanisms, such as Snort IDS, which can be deployed on edge devices with minimal latency. Instead of performing exhaustive system-wide scans, the methodology prioritizes selective security scanning, focusing on high-risk components to reduce processing overhead while maintaining a robust security posture. Moreover, the distributed nature of SESAME's security monitoring allows certain security computations to be offloaded to external monitoring units or cloud-based security services when necessary, preventing undue strain on the robot's onboard computing resources.

6.2. Future Extensions

As robotic systems continue to evolve and expand into diverse operational domains, future extensions of the SESAME security framework must focus on enhanced automation, AI-driven risk assessment, and cross-domain applicability. While the current methodology provides a strong foundation for security assessment in MRS, incorporating AI and ML can significantly improve risk modeling, anomaly detection, and adaptive threat response mechanisms.

One of the key areas for future development is the integration of AI and ML algorithms to further automate risk modeling and assessment processes. Currently, SESAME relies on structured security repositories (i.e., CVE, CWE, and CAPEC) to map vulnerabilities to potential attack paths. However, this process can be enhanced through AI-driven analysis, where models trained on historical cyber incidents, attack patterns, and real-time security data can achieve the following:

- Predict emerging threats based on observed vulnerabilities and evolving attack techniques;
- Automatically generate and update Template Attack Trees based on live threat intelligence feeds;
- Enhance anomaly detection by training ML models to recognize deviations in robotic behavior that may indicate security breaches;

- Optimize security EDDIs by enabling self-learning response mechanisms, ensuring that robotic systems adapt dynamically to new threats without human intervention.

By leveraging deep learning models and reinforcement learning, SESAME can evolve from a reactive security framework into a proactive and adaptive risk assessment solution.

While SESAME has been successfully deployed in critical infrastructure environments such as power stations and healthcare robotics, its methodology can be extended to other high-risk domains where cybersecurity is essential. One particularly promising area for SESAME's application is precision agriculture, where autonomous drones are increasingly used for monitoring crops, soil conditions, and livestock. These drones rely heavily on GPS navigation, wireless communication, and sensor data, making them vulnerable to GPS spoofing, unauthorized access, and data interception. SESAME can be adapted to enhance UAV security by integrating real-time intrusion monitoring, GPS integrity verification, and encrypted communication protocols. Furthermore, machine learning-based threat detection could analyze drone flight patterns and sensor anomalies to identify potential cyber intrusions or manipulations, triggering automatic security responses to protect agricultural assets.

Another significant domain for SESAME's expansion is autonomous transportation and logistics, where self-driving vehicles and robotic fleets are revolutionizing supply chain management. These systems, however, are exposed to cyber threats such as remote hacking, sensor spoofing, and communication disruptions, which could compromise route optimization, cargo security, and operational efficiency. SESAME's methodology can be applied to secure vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications, ensuring encrypted data exchange and intrusion-resistant navigation systems. By extending security EDDIs to this domain, SESAME can detect unauthorized network access, MITM attacks, and cyber-induced anomalies in vehicle behavior, triggering automated countermeasures to mitigate risks in real time.

Beyond industrial applications, SESAME's framework can also contribute to the security of smart cities and public safety robotics. With the increasing deployment of surveillance drones, autonomous emergency response robots, and interconnected IoT-based systems, smart cities face complex cybersecurity challenges. Attackers could manipulate these robotic systems to disrupt public safety operations, compromise surveillance feeds, or exploit weak authentication mechanisms. By adapting SESAME's attack trees and risk assessment models, urban robotics can be equipped with secure communication protocols, anomaly detection mechanisms, and automated security responses. This would ensure the data integrity, operational reliability, and resilience of interconnected robotic networks in smart city environments.

By expanding into these emerging domains, SESAME can provide a scalable and adaptive security framework that strengthens cyber resilience across autonomous and CPS. Through AI-driven threat modeling, real-time risk assessment, and automated mitigation strategies, SESAME has the potential to enhance security in agriculture, transportation, logistics, and public safety, ensuring the reliable and secure deployment of autonomous systems in diverse operational environments.

7. Conclusions

The SESAME project introduces a robust methodology for the automated security assessment of MRS. By leveraging repositories like CVE, CWE, and CAPEC along with tools such as OpenVAS, the methodology effectively identifies vulnerabilities, maps them to potential attacks, and develops mitigation strategies. Template Attack Trees and EDDIs enhance real-time monitoring and management. Successful applications in power stations and healthcare settings demonstrate the methodology's effectiveness in managing security

risks. This work highlights the importance of a structured approach to improve the resilience of robotic systems against cyber–physical threats.

Future efforts should focus on expanding vulnerability repositories to include new threats, enhancing automation in attack tree generation, and integrating machine learning for better threat prediction. Extending the methodology to various robotic platforms and environments will validate its versatility. Collaboration between academia, industry, and regulatory bodies is essential to develop standardized security frameworks, ensuring the safe and secure operation of robotic systems in an interconnected world. Also, promising approaches are trying to make use of AI to further enhance the automation of security modelling and assessment. The application of SESAME in other application domains, such as in precision agriculture with the use of drones or in industrial settings, is also of interest.

Author Contributions: Conceptualization, M.P., E.M. and G.H.; methodology, M.P., E.M., M.M., A.S., P.N., E.S. and G.B.; software, M.P. and E.M.; validation, M.M., A.S., P.N., E.S. and G.B.; writing—original draft preparation, G.H. and M.P.; writing—review and editing, G.H.; supervision, S.I. All authors have read and agreed to the published version of the manuscript.

Funding: This work has received funding from the European Union’s Horizon 2020 research and innovation programs under grant agreements No. 101017258 (SESAME), No. 101128070 (CONSOLE), No. 101159751 (PATTERN), and No. 101070599 (SecOPERA).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Author Gizem Bozdemir was employed by the company PAL Robotics SL. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Yaacoub, J.-P.A.; Noura, H.N.; Salman, O.; Chehab, A. Robotics cyber security: Vulnerabilities, attacks, countermeasures, and recommendations. *Int. J. Inf. Secur.* **2022**, *21*, 115–158. [[CrossRef](#)] [[PubMed](#)]
2. Fournaris, A.P.; Tselios, C.; Haleplidis, E.; Athanasopoulos, E.; Dionysiou, A.; Mitropoulos, D. Providing Security Assurance & Hardening for Open Source Software/Hardware: The SecOPERA approach. In Proceedings of the 2023 IEEE 28th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Edinburgh, UK, 6–8 November 2023; pp. 80–86.
3. White, R.; Caiazza, G.; Christensen, H.; Cortesi, A. SROS1: Using and developing secure ROS1 systems. In *Robot Operating Systems (ROS)*; SCI; Springer: Cham, Switzerland, 2018; Volume 778, pp. 373–405.
4. Quigley, M.; Conley, K.; Gerkey, B.P.; Faust, J. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009; Volume 3.2, pp. 5–10.
5. McClean, J.; Stull, C.; Farrar, C.; Mascareñas, D. A Preliminary Cyber-Physical Security Assessment of the Robot Operating System (ROS). In Proceedings of the Unmanned Systems Technology XV, Baltimore, MD, USA, 29 April–3 May 2013; Volume 8741, p. 874110.
6. Quarta, D.; Pogliani, M.; Polino, M.; Maggi, F.; Zanchettin, A.M.; Zanero, S. An Experimental Security Analysis of an Industrial Robot Controller. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 268–286.
7. Hatzivasilis, G.; Soultatos, O.; Anicic, D.; Bröring, A.; Fysarakis, K.; Spanoudakis, G. Secure Semantic Interoperability for IoT Applications with Linked Data. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–7.
8. *ISO 10218-2:2025; Robotics—Safety Requirements—Part 2: Industrial Robot Applications and Robot Cells*. International Organization for Standardization (ISO): Geneva, Switzerland, 2025.
9. Hollerer, S.; Fischer, C.; Brenner, B.; Papa, M.; Schlund, S.; Kastner, W.; Fabini, J.; Zseby, T. Cobot attack: A security assessment exemplified by a specific collaborative robot. *Procedia Manuf.* **2021**, *54*, 191–196. [[CrossRef](#)]
10. Deng, G.; Zhou, Y.; Xu, Y.; Zhang, T.; Liu, Y. An investigation of byzantine threats in multi-robot systems. In Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), San Sebastian, Spain, 6–8 October 2021; pp. 17–32.

11. Global Times. Mainframe Malfunction Causes Dozens of Drones to Crash into Building in SW China. 2021. Available online: <https://www.globaltimes.cn/page/202101/1214165.shtml> (accessed on 16 July 2024).
12. Koscher, K.; Czeskis, A.; Roesner, F.; Patel, S.; Kohno, T.; Checkoway, S. Experimental security analysis of a modern automobile. In Proceedings of the 2010 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 16–19 May 2010; pp. 447–462.
13. Vasconcelos, G.; Miani, R.S.; Guizilini, V.; Souza, J.R. Evaluation of DoS attacks on commercial Wi-Fi-based UAVs. *Int. J. Comput. Netw. Inf. Secur.* **2019**, *11*, 212–223. [CrossRef]
14. Xu, Y.; Deng, G.; Zhang, T.; Qiu, H.; Bao, Y. Novel denial-of-service attacks against cloud-based multi-robot systems. *Inf. Sci.* **2021**, *576*, 329–344. [CrossRef]
15. Giaretta, A.; De Donno, M.; Dragoni, N. Adding salt to pepper: A structured security assessment over a humanoid robot. In Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES), Hamburg, Germany, 27–30 August 2018; Volume 22, pp. 1–8.
16. Ashibani, Y.; Mahmoud, Q.H. Cyber physical systems security: Analysis, challenges and solutions. *Comput. Secur.* **2017**, *68*, 81–97. [CrossRef]
17. Wynn, J. Threat Assessment and Remediation Analysis (TARA), MITRE. 2014. Available online: <https://www.mitre.org/sites/default/files/2021-10/pr-14-2359-tara-introduction-and-overview.pdf> (accessed on 16 July 2024).
18. Hatzivasilis, G.; Lakka, E.; Athanatos, M.; Ioannidis, S.; Kalogiannis, G.; Chatzimpyros, M.; Spanoudakis, G.; Papastergiou, S.; Karagiannis, S.; Alexopoulos, A.; et al. Swarm-intelligence for the moder ICT ecosystem. *Int. J. Inf. Secur.* **2024**, *23*, 2951–2975. [CrossRef]
19. MITRE. Common Vulnerabilities and Exposures. Available online: <https://www.cve.org/> (accessed on 16 July 2024).
20. NIST. National Vulnerability Database. Available online: <https://nvd.nist.gov/> (accessed on 16 July 2024).
21. MITRE. Common Weakness Enumeration. Available online: <https://cwe.mitre.org/> (accessed on 16 July 2024).
22. MITRE. Common Attack Pattern Enumerations and Classifications. Available online: <https://capec.mitre.org/> (accessed on 16 July 2024).
23. Vilches, V.M.; Usategui, L.; Izquierdo, R. Introducing the Robot Vulnerability Database (rvd), Aliasrobotics. 2021. Available online: <https://github.com/aliasrobotics/RVD> (accessed on 16 July 2024).
24. Settanni, F.; Regano, L.; Basile, C.; Lioy, A. A model for automated cybersecurity threat remediation and sharing. In Proceedings of the 2023 IEEE 9th International Conference on Network Softwarization (NetSoft), Madrid, Spain, 19–23 June 2023; pp. 492–497.
25. Schauer, S.; Polemi, N.; Mouratidis, H. MITIGATE: A dynamic supply chain cyber risk assessment methodology. *J. Transp. Secur.* **2019**, *12*, 1–35. [CrossRef]
26. Raicu, G.; Raicu, A. Cybersecurity strategy in industry 4.0. *Int. J. Mod. Manuf. Technol.* **2022**, *14*, 233–238. [CrossRef]
27. Basan, A.; Basan, E. The Methodology for Assessing Information Security Risks for Robotic Systems. In Proceedings of the 4th International Conference on “Information Technology and Nanotechnology” (ITNT-2020), Samara, Russia, 26–29 May 2020; Volume 2667, pp. 30–35.
28. Kenta, K.; Washizaki, H.; Fukazawa, Y.; Ogata, S.; Okubo, T.; Kato, T.; Kanuka, H.; Hazeyama, A.; Yoshioka, N. Tracing CAPEC attack patterns from CVE vulnerability information using natural language processing technique. In Proceedings of the 54th Hawaii International Conference on System Sciences, Maui, HI, USA, 5–8 January 2021; pp. 6996–7004.
29. PILZ. White Paper Security. 2018. Available online: https://www.pilz.com/mam/pilz/content/uploads/wp_security_en_2018_10.pdf (accessed on 16 July 2024).
30. OpenVas. Available online: <https://www.greenbone.net/en/testnow/#toggle-id-1> (accessed on 16 July 2024).
31. Snort. Available online: <https://www.snort.org/> (accessed on 16 July 2024).
32. Aslansefat, K.; Nikolaou, P.; Walker, M.; Akram, M.N.; Sorokos, I.; Reich, J.; Kolios, P.; Michael, M.K.; Theocharides, T.; Ellinas, G.; et al. SafeDrones: Real-Time Reliability Evaluation of UAVs Using Executable Digital Dependable Identities. In *Model-Based Safety and Assessment (IMBSA)*; LNCS; Springer: Cham, Switzerland, 2022; Volume 13525, pp. 252–266.
33. Tang, G.; Webb, P. Human–robot shared workspace in aerospace factories. In *Human–Robot Interaction*; Taylor & Francis: Abingdon, UK, 2019; pp. 72–79.
34. Nikolaou, P.; Savva, A.; Sorokos, I.; Aslansefat, K.; Missaoui, S.; Naveed, A.; Hillen, D.; Lorenz, M.; Walker, M.; Papoutsakis, M.; et al. Safe, Secure and Dependable Multi-UAV Systems for Search and Rescue Operations. In Proceedings of the Design, Automation and Test in Europe Conference (DATE), Lyon, France, 31 March–2 April 2025; pp. 1–10.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.