TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Joosep Parts 221963IVCM

ATTACK PATTERN ASSESSMENT OF TELEPRESENCE ROBOTS IN HEALTHCARE SYSTEMS CONTEXT

Master's Thesis

Supervisor: Kaido Kikkas
Ph.D.

Co-supervisor: Janika Leoste
Ph.D.

Tallinn 2024

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Joosep Parts 221963IVCM

KAUGOSALUSROBOTITE RÜNDEMUSTRITE ANALÜÜS TERVISHOIUSÜSTEEMIS

Magistritöö

Juhendaja: Kaido Kikkas
Ph.D.

Kaasjuhendaja: Janika Leoste
Ph.D.

Tallinn 2024

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Joosep Parts

12.05.2024

Abstract

Telepresence robots are increasingly employed in healthcare systems, enhancing remote patient care capabilities. Yet, the adoption of such advanced technologies also broadens the potential cyber(physical) attack surface, introducing risks previously unconsidered. Ensuring security within robotics systems is inherently difficult due to the complexity of these systems, contributing to a potential security gap. To address this gap, a combined threat modeling approach based on Attack Pattern analysis assisted by Large Language Models was employed. This method unveiled 312 potential attack paths in the common telepresence robot system. By using Natural Language Processing techniques, such as sentiment analysis for context assessment, semantic similarity analysis with Universal Sentence Encoder for scenario validation, and Natural Language Generation for threat scenario description, the research provides a unique approach to threat modeling. The findings are represented in an attack tree format, where attack paths are weighted based on metrics like severity and likelihood, and scenario descriptions are made readable even to the non-technical audience. The computer-generated scenarios show a promising level of similarity of 0.670 compared to participant generated scenarios, indicating the potential of AI-assisted threat modeling. The majority of potential attacks are likely to originate from the software domain, accounting for 57% of identified attack paths, suggesting a prioritization in securing this area, while also addressing significant threats from hardware (15%) and supply chain (14%) domains, and not overlooking social engineering (8%) and physical security (6%). Additionally, findings reveal that there exists a possibility of 64% that an adversary might impact healthcare system's software, 28% for hardware and only 8% chance for impacting the personnel. The presented attack tree with possible scenarios could help cybersecurity professionals to make better informed decisions when it comes to securing telepresence robot system in healthcare systems in the future.

The thesis is written in English language and is 77 pages long, including 5 chapters, 27 figures and 8 tables.

List of Abbreviations and Terms

BI	Business Intelligence
CAPEC	Common Attack Pattern Enumeration and Classification
CMC	Computer-mediated communication
COTS	Commercial Off The Shelf
CPS	Cyber-physical system
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
DAN	Deep Averaging Network
DoS	Denial of Service
ENISA	European Union Agency for Cybersecurity
EoP	Elevation of Privilege
FBI	Federal Bureau of Investigation
GRL	Goal-oriented Requirement Language
HCS	Healthcare system
ICS	Industrial Control Systems
ICT	Information and Communication Technology
IoT	Internet of Things
LLM	Large Language Model
MAL	Meta Attack Language
NLG	Natural Language Generation
NLP	Natural Language Processing
ROS	Robotic Operating System
SRPS	Social Robots in Public Spaces
SSA	Semantic Similarity Analysis
T-MAP	Threat Modeling Attack Path Analysis
TP	Telepresence
TPR	Telepresence robot
TTP	Tactics, Techniques and Procedures
USE	Universal Sentence Encoder
VE	Virtual enviorment
VR	Virtual reality
WEF	World Economic Forum

Table of Contents

List of Figures

List of Tables

1. Introduction

The rapid growth of technology and robotics has led to significant advancements in Information and Communication Technology (ICT) infrastructure worldwide. Over the last decade, integration of physical and cybernetic parts via networks (also known as Industry 4.0), has brought opportunities, but also new challenges for Internet of Things (IoT) devices and other Cyber-physical system (CPS) [1]. This evolution of technology has boosted the field of robotics, resulting in a wide array of potential applications for robotics in the near future for Healthcare system (HCS) [2, 3]. Aside from manufacturing, agriculture and transport, healthcare robotics sector is one of the most prominent sectors [4], with some estimates suggesting that the healthcare robotics market will grow from $16.8 billion in 2024 to $44.4 billion by 2030 [5]. Various applications for healthcare robotics are already being used in practice, such as elderly care, surgical assistants, medical transport among other similar applications [6]. One such robot which stands out due to its versatility is a Telepresence robot (TPR). TPR is a remote-controlled, wheeled device equipped with a camera, microphone, and display to facilitate communication and interaction between people in different locations, often used to support patients and elderly individuals by providing virtual presence and assistance [7]. Preliminary findings indicate that TPRs offer promising solutions for healthcare professionals and patients, particularly in situations where physical presence is impossible or isolation is required to prevent contagion [8]. Some examples of TPRs are shown in Figure 1 divided into two classes: commercial and healthcare/social care specific robots. TPRs in combination with new technologies such

Double3, temi, Ohmni, PadBot P2, Awabot

(a) Examples of commercial telepresence robots

Pepper, InTouch, ARI, SIFROBOT, RAMCIP

(b) Examples of healthcare telepresence robots

Figure 1. Examples of different kinds of telepresence capable robots.

as Virtual reality (VR), Computer-mediated communication (CMC), Virtual enviorment (VE) and Telepresence (TP) are changing the dynamics of how traditional healthcare is being provided and will likely improve quality of care [9]. Although the advantages of

TPRs in HCS are numerous, this technology also creates new possible security risks that need to be assessed.

According to Federal Bureau of Investigation (FBI) 2023 Internet Crime Report, health sector is the most affected sector by cybercrime (249 reports) in U.S. with total potential losses exceeding $12.5 billion [10]. It has been also suggested, that the link between cybersecurity, robotics and regulatory compliance is weak [11], which could potentially lead to a situations where the HCS is vulnerable to cyberattacks, as the usage of robotic systems increases. Rising cyber threats in HCS [10], the increasing use of robotic applications in the HCS domain [9] and the immature state of cybersecurity in robotics [12], all contribute to the growing challenges in securing robotic systems. Therefore, effective cybersecurity measures are essential to prevent unauthorized access and breaches, which could result in identity theft, privacy violations, and misuse of medical information [13, 14]. However, the research regarding cybersecurity in TPR systems is limited, and the existing research mostly focuses on the functionality and usability of the TPR systems [15, 16], rather than on the cybersecurity aspects.

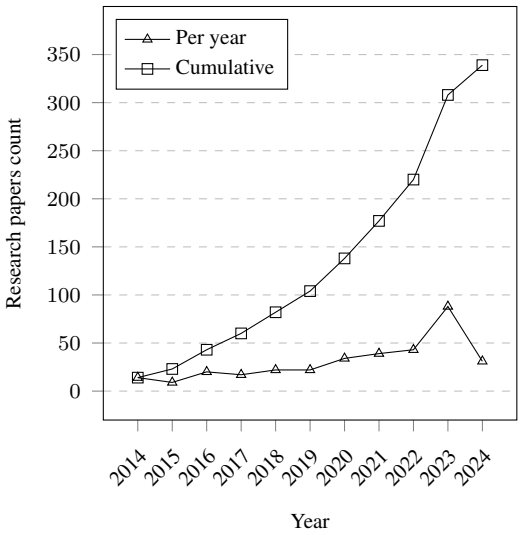
1.1 Problem statement

It is inherently difficult to ensure security within robotics systems due to the complexity of robotic systems in general, leading to wide attack surfaces and a variety of potential attack vectors [12, p. 2]. Institutions should be aware that by introducing new technologies, such as the TPR technology, they might also inadvertently introduce new possible security risks which come with the technology [17, 18]. And TPR are no exception to this, as they are also susceptible to various cybersecurity issues either hardware, firmware or application layer [19]. A newer form of cybersecurity threats originate from CPS domain, where the cyber and physical worlds are tightly integrated and robots are given freedom to navigate and interact with the physical world [20, 21]. CPS threats could extend the range of known attack vectors by utilizing robotic capabilities and creating new attack surfaces not considered before [22]. In addition, robotics manufacturers often struggle to mitigate vulnerabilities in reasonable time periods [12], which leaves the systems vulnerable to exploitation. To combat this, institutions deploying Robotic Operating System (ROS) devices tend to use perimeter countermeasures (51%) and network segmentation and segregation (48%) as their mitigation strategy [12]. The lack of investment in cybersecurity and the immature state of the field in robotics cybersecurity contribute to the challenges in securing robotic systems and defensive approaches are struggling to keep up with the need for security [12]. The security of robotic systems is continually challenged by the complexity and evolving nature of the technologies, despite efforts to mitigate these risks through common cybersecurity strategies.

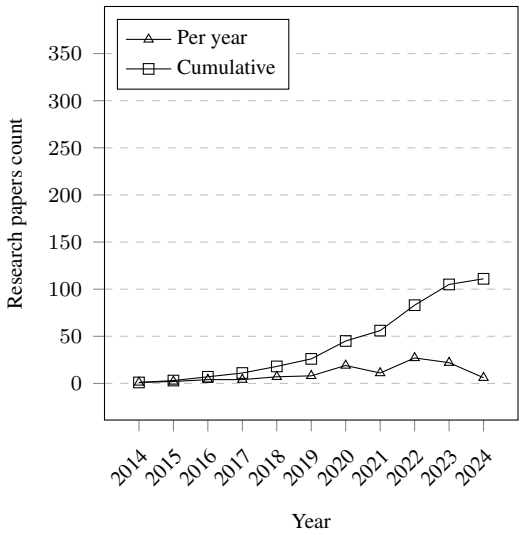
Though there exists various risk assessment models, frameworks, and methodologies to assess cybersecurity risks within robotics systems in general, the studies which focus on TPR are limited. State-of-the-art knowledge on robotic systems cybersecurity in general, comes from the Industrial Control Systems (ICS) domain or from IoT devices related research [12, 17]. However, since TPRs are consumer oriented devices, and their usage in healthcare settings and it's applications are still being explored [15, 16], it is difficult to apply the similar risk assessment models which can be found from ICS and IoT domain on a much smaller scale, such as the TPR system. Since the use of TPR technology is increasing and there will always be new emerging cybersecurity threats, there exists a need to explore the possible cybersecurity weaknesses in TPR systems.

1.2 Motivation

It is known that cybersecurity related research in HCS often times lags behind the development of new technologies [23]. Research focusing on TPRs can also be seen in Figure 2 as the number of publications in the related field is on the rise and increased significantly in past few years 2020–2024. Figure 2 shows articles and conference papers focusing on healthcare systems cybersecurity and the use of TPR for past 10 years. At the moment of writing, only partial data is available for 2024. Recent events, such as the COVID-19 pandemic, has expedited the adaptation of robotic technology, including TPR technology into people's lives [24, 25]. However, there still exists nearly 3 times less cybersecurity research papers on the topic, which is concerning.



(a) TPRs research papers related to HCS context. Queried from IEEExplore [26], Scopus [27], Springer [28] and Web of Science [29].



(b) TPRs cybersecurity research papers. Queried from IEEExplore [30], Scopus [31], Springer [32] and Web of Science [33].

Figure 2. Research paper trends of telepresence robotics.

Cybersecurity is crucial in HCS as data which can be compromised is sensitive information, consisting of personal and medical data [34]. HCSs are attractive targets for cybercriminals for ransomware attacks as attacker motives are financial gain (98%) according to Verizon's 2023 Data Breach Investigations Report [34]. In Europe, health sector related attack vectors are of type: 54% ransomware, 46% data related, 13% intrusion attacks, with a median cost of a major security incident reported to be around $300.000 euros [35]. In addition, 2023 World Economic Forum (WEF) Global Risks Report indicates that *Widespread cybercrime and cyber insecurity*, is currently 8th global risk ranked by severity and will remain so in the long term (10 years) [36]. Survey on robots cybersecurity, reveals that robotic systems can be attacked in various creative ways, either through ROS, physically, through the network or some other way, impacting the robotic system's integrity, availability, and confidentiality [20]. These indicators show that cybersecurity should be a relevant topic in HCS and this trend is unlikely to change in the near future. Due diligence should be carried out to identify and mitigate possible cybersecurity risks in third-party relationship TPR systems in HCS before they actually materialize. To respond to this requirement, an understanding of the potential issues and the mechanisms through which they might arise is necessary.

One way to answer such complex cybersecurity related questions is by applying threat modeling methodologies to describe cyber(physical) threats [37]. To explore the application of threat modeling within cybersecurity, the systematic literature review conducted by Xiong and Lagerström in 2019 offers valuable insights. It categorizes relevant literature into three primary domains: the application of threat modeling, methods of threat modeling, and the threat modeling process itself. The review identified 29 articles focused on how to apply threat modeling, 20 articles detailing various threat modeling methods, and 5 articles outlining the threat modeling process [38]. This shows that threat modeling is a complex dynamic field with a lot of different methodologies and processes. Researchers use standardized methodologies in their work when applicable, for example by using descriptive language Meta Attack Language (MAL) [39] or using Structured Threat Information (STIX) [40] to describe threat scenarios. But as threat modeling is a diverse field lacking common ground [38], it can prove to be a difficult to apply existing threat modeling methodologies to new technologies as the context is not quite the same. Thus, each new system, technology or application under assessment, requires a tailored threat modeling methodology to be developed, building on top of the existing knowledge.

In the context of threat modeling and generating cyber-physical attack paths, as suggested by Stellios *et al.*, by utilizing Natural Language Processing (NLP) we could get better environmental scores and the characteristics of threat agents [41]. Stellios *et al.*, however did not attempt to use this approach, but it is a valid approach to consider. Thus, we will

focus on creating a method which allows us to generate and visualize cyber-physical attack paths within TPR systems when viewed from the attackers' perspective by utilizing NLP and Semantic Similarity Analysis (SSA). Initial steps of threat modeling process in the given thesis will align with the methodology utilized by Stellios *et al.* which was used to assess IoT devices against critical systems. But we will focus on TPR systems in HCS context, and we will use Natural Language Processing (NLP) with SSA to validate the generated attack paths. As a last step, we visualize the generated attack paths in a node graph format (attack tree), which will allow us to better understand the output of the model and use it to assess found attack patterns.

1.3 Scope and goal

The goal of this research is to develop a method for generating and visualizing cyber(physical) attack paths within TPR systems in the HCS context. The study is aimed at highlighting potential attack paths, in attack tree format using TPR components as attack vectors. The goal can be best described in Goal-oriented Requirement Language (GRL) style when viewed from the adversary's perspective (see Figure 3).

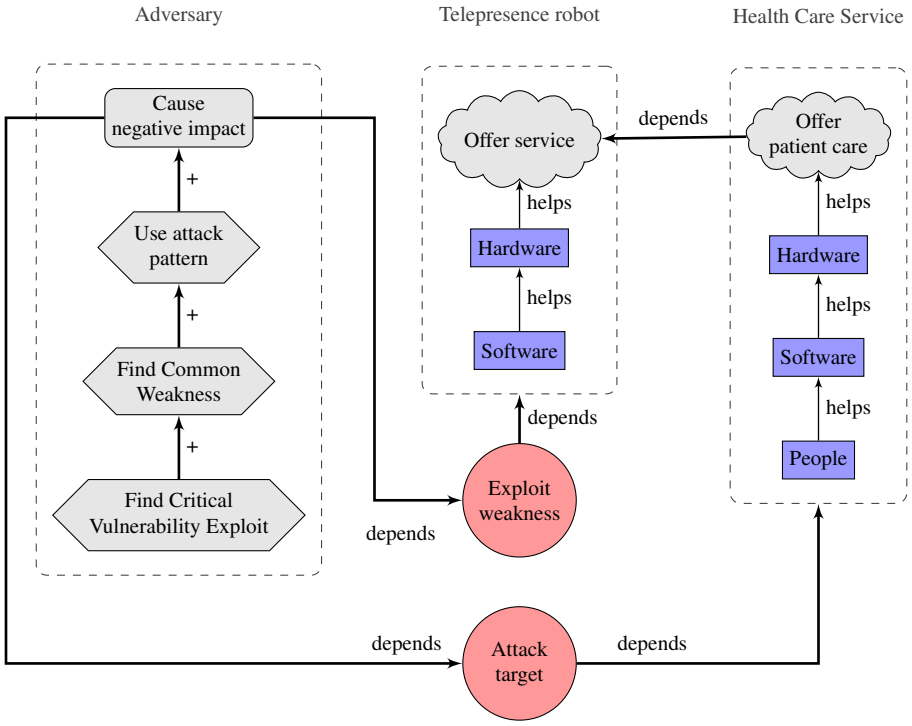


Figure 3. Research scope and focus.

The goal of the adversary is to attack and cause negative impact on the HCS by exploiting a weakness in TPR system service on which the HCS patient care service quality depends on. To this end, the study will focus on exploring: 1) how the adversary can attack and cause

negative impact to HCS; 2) which patterns, weaknesses can be exploited. Main outcome will be a graphical attack tree model with scenario descriptions which was generated by the proposed method and validated by the semantic text based validation process using the user-generated scenarios as a reference. This high level overview of various avenues of approach can be used to inform security decisions and to develop future mitigation strategies when deploying TPR systems.

1.4 Research questions

The central research problem tackled in this thesis explores cyber(physical) risks introduced by the use of TPR within HCS. Specifically, it seeks to investigate the potential cyber(physical) attack paths that could be exploited by malicious actors, and the subsequent impact these could have on the HCS. The research questions are:

RQ1: What are the potential cyber(physical) attack paths that could be exploited through TPR systems?
RQ2: What are the choke points and high value targets of the system?

2. Background

In more recent events, COVID-19 has expedited the adaptation of robotic technology in HCS [42], including TPR technology into people's lives [24] and the use of robotics in HCS is increasing, as TPR are being accepted in HCS [43]. Practical experimentation with the devices report that in general, TPRs are pleasant and practical devices to use [44]. The lessons learned from the pandemic will likely lead to a more widespread use and development of TPR technology in the future, such as development of *FLEXTRA* shows, that TPR are being specifically designed for HCS [45, 46]. However, the majority of research still focuses on the usability and functionality of using consumer oriented devices in HCS, rather than the development of dedicated devices. TPRs have great potential for medical reasons within HCS as they benefit personnel the replacement of physical presence and provide access to medical specialists for the patients in restricting environments (COVID, remote location) [44, 47]. A typical TPR consists of several hardware components and

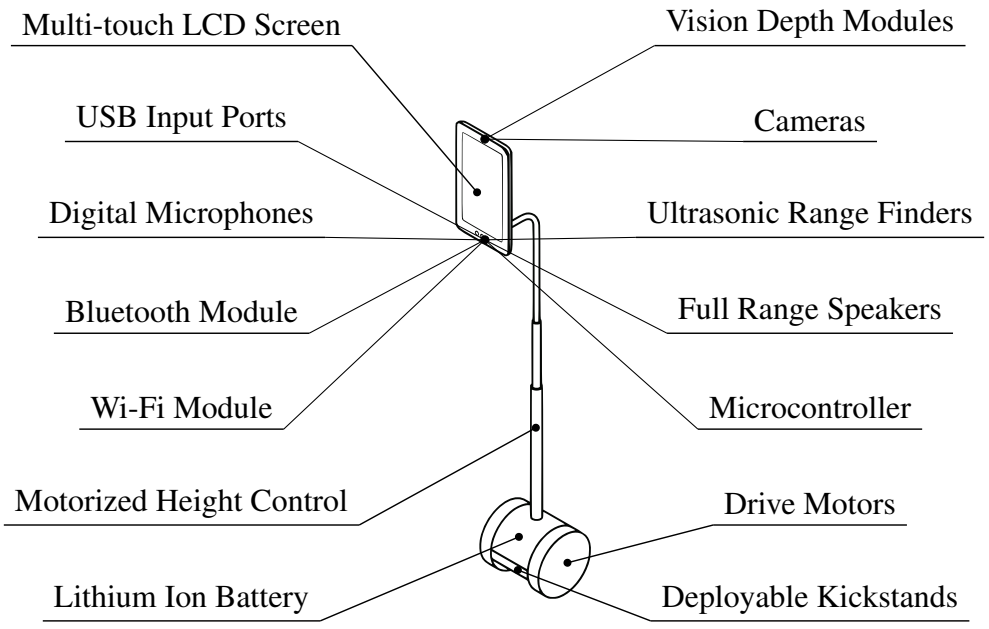


Figure 4. Common telepresence robot's hardware components and I/O interfaces.

I/O interfaces. The features and components can vary between different manufacturers, however the devices are often equipped with similar features and components. Figure 4 depicts the hardware components and I/O interfaces of a fictional (yet realistic) TPR. The generalized selection of components is derived from the most common features and components found from TPRs as listed in Table 1.

Table 1. Component features in various commercial telepresence capable robots.

	Double3[1]	temi[2]	Ohmni[3]	PadBot P2[4]	Awabot[5]	Pepper[6]	InTouch[7]	ARI[8]	SIFROBOT[9]	RAMCIP[10]
Camera	X	X	X	X	X	X	X	X	X	X
Microphones	X	X	X	X	X	X	X	X	X	X
Ultrasonic Range Finders	X		X				X			X
Full Range Speakers	X	X	X	X	X	X	X	X	X	X
Microcontroller	X	X	X	X	X	X	X	X	X	X
Drive Motors	X	X	X	X	X	X	X	X	X	X
Deployable Kickstands	X									
Multi-touch LCD Screen	X	X	X	X	X	X	X	X	X	X
USB Input Ports	X	X		X	X	X	X	X		
Digital Microphones	X	X	X	X	X	X	X	X	X	X
Bluetooth Module	X	X	X			X	X			
Wi-Fi Module	X	X	X	X	X	X	X	X	X	X
Motorized Height Control	X			X		X		X		
Lithium Ion Battery	X	X	X	X	X	X	X	X	X	X

Focusing research on the components and I/O interfaces of the TPR which are common for the devices, we hope to identify possible attack vectors and paths which are common for the devices. Research on TPR security is scarce, as the focus has primarily been on the usability and functionality of the devices, rather than the security implications of using them. Thus, we aim to bridge the gap in the literature by providing a novel approach to assessing cybersecurity risks in TPRs by threat modeling and generating possible cyber(physical) attack paths for TPRs by focusing on the common components and I/O interfaces as attack vectors. The following sections describe the terminology, cybersecurity risks in TPR, threat modeling, attack patterns, and the use of Large Language Models (LLMs) for NLP and Natural Language Generation (NLG) tasks which will aid in the threat modeling process.

[1]https://www.doublerobotics.com/double3.html
[2]https://www.robotemi.com/specs/
[3]https://ohmnilabs.com/products/customers/faq/spec
[4]https://www.roboserv-solutions.com/p2
[5]https://telepresence.awabot.com/produit/beam-pro
[6]https://www.wevolver.com/specs/softbank-robotics-pepper
[7]https://robotsguide.com/robots/vita
[8]https://spring-h2020.eu/news/spring-ari-robot-specifications
[9]https://sifrobot.com/product/robot-sifrobot-4-2
[10]https://cordis.europa.eu/project/id/643433

2.1 Cybersecurity risks in telepresence robots

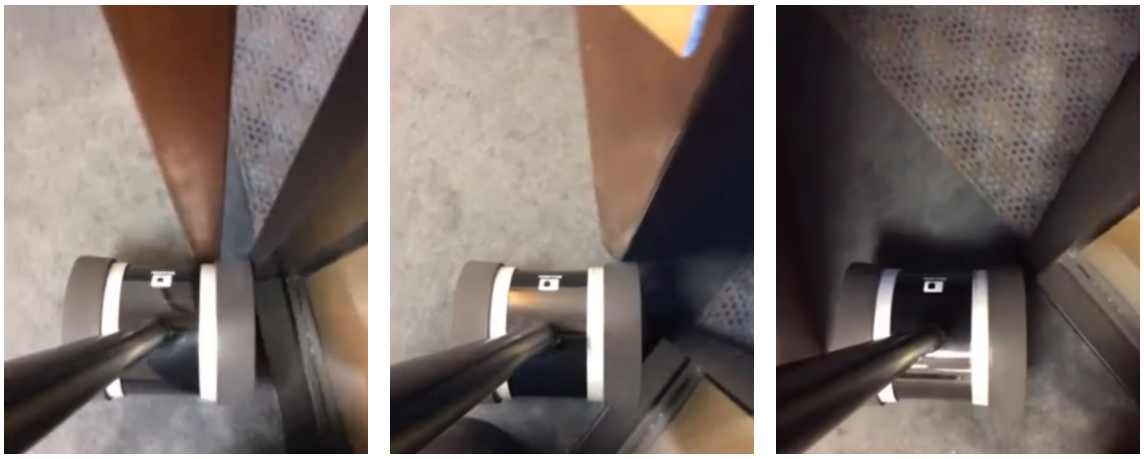
Robots and autonomous systems are facing cybersecurity problems similar to computers. This includes concerns for critical tasks by surgical or military robots, as well as household robots like vacuum cleaners, which can impact the privacy and safety of users if compromised [48]. In the field of robotics, cyber-attacks such as Denial of Service (DoS), spoofing, and man-in-the-middle attacks are common [49]. Some experts identify the primary threats as remote access, spying, and eavesdropping, with network connectivity being a significant vulnerability [50]. Morante *et al.* points out the lack of encryption and authentication in current robotic communication protocols is a common problem [51]. Cybersecurity discussions surrounding care robots underscore the convergence of cyber-physical vulnerabilities, network security risks, data privacy concerns, regulatory deficiencies [52]. The review by Oruma *et al.* delves into the multifaceted issues of security and privacy associated with social robots, considering a wide array of perspectives from different stakeholders [53, 54]. The study emphasizes the urgent need for tailored security standards and frameworks specifically for Social Robots in Public Spaces (SRPS), to maintain secure and ethical operations. Such measures are crucial to safeguard individual rights and public safety, facilitating the smooth integration of these robots into society [53]. Despite these common shortcomings, cybersecurity risks in TPR are not well documented in the literature.

Table 2. Examples of TPR vulnerabilities.

Vulnerability	**Short description**
Physical Interaction [55]	Attacker uses Double 2 TPR to open doors, accessing restricted areas.
Unauthorized Access [56, 57]	Active developer tool in VGo Celia allows access to pictures and video feeds.
API Manipulation [58, 59]	Exploitation of Double Robotics telepresence robots' API for unauthorized access and control.
Data Interception [60]	Intercepted firmware updates from VGo Celia leads to potential theft of sensitive data and unauthorized access to video recordings.
Ransomware [61]	Attacker uses ransomware to control hospital robots, Pepper and NAO for espionage or disruption.
Remote Hijacking [62]	Ethical hackers demonstrated the potential for remote control of robotic actions on Alpha 2, showcasing a compromised robot's ability to perform hazardous tasks like stabbing.

Alternative sources do however expose potential risks in TPR technology (see Table 2). In a demonstration by Tweedian, as seen in Figure 5, attacker leverages Double 2 TPR physical body to interact with the environment to open a door and gain access to restricted

areas [55]. In Vecna VGo Celia case, a developer tool was left active on the robot that made



(a) Attacker connects to robot and finds a door partially open.

(b) Attacker uses robot's body to push the door open further.

(c) Door is now fully open to previously restricted area.

Figure 5. Example of a cyber-physical risk.

the device vulnerable to unauthorized access to pictures and video feeds [56]. Heiland identified vulnerabilities in Double Robotics telepresence robots which allowed attacker with physical access to the device, to exploit vulnerabilities that allowed unauthorized access to device information and control, including serial numbers, GPS coordinates, and user tokens, through manipulated API requests and intercepted static user tokens [58]. Prior to public disclosure of the vulnerabilities, the manufacturers were notified and the vulnerabilities were patched for all cases. However, such incidents highlight both the potential risks and creativity of attackers in exploiting vulnerabilities in TPR technology.

2.2 Related work

Hankin, Malacaria, *et al.* created an automatic attack graph generation framework called Attack Dynamics, leveraging Common Attack Pattern Enumeration and Classification (CAPEC), Common Weakness Enumeration (CWE) and Common Vulnerabilities and Exposures (CVE) databases to generate detailed visualizations of potential attack paths [63]. This tool facilitates the automated enumeration of vulnerable scenarios, links enterprise mitigations to security flaws, and integrates optimization tools for cost-effective security measures [63]. Notably, it serves as a what-if analysis tool for various network configurations and access privileges, presenting a significant stride in threat modeling for cybersecurity [63]. The paper also alludes to future expansions, including an advanced visualization system and automated attack execution, which could greatly enhance the capability to simulate and understand complex cyberattacks [63].

The paper by Stellios *et al.* presents a novel risk-based methodology for identifying and assessing IoT-enabled cyber-physical attack paths against critical systems. Unlike traditional approaches that focus on cyber system connectivity, this methodology considers both cyber and physical interactions and employs an attack tree topology to efficiently model potential attack scenarios. The approach is distinct in its use of CVE and Common Vulnerability Scoring System (CVSS) like metrics for threat modeling to prioritize identified attack paths, thereby reducing false positives and aiding decision-makers in effectively mitigating risks [41]. The study's validation, using a healthcare scenario, demonstrates its efficacy in identifying and evaluating hidden and complex attack paths that may have been previously underestimated [41].

Brazhuk explored the challenges of building and utilizing a formal knowledge base, integrating ATTCK, CAPEC, CWE, and CVE security enumerations for threat modeling [64]. The proposed model, constructed as an ontology in OWL (Web Ontology Language) and RDF (Resource Description Framework) formats, aids in understanding the relationships between attack techniques, patterns, and system vulnerabilities, thus assisting in constructing various threat landscapes [64]. This model facilitates automated threat modeling, providing valuable insights for security analysis [64].

2.3 Threat modeling concepts

In order to provide the reader with an understanding of the terminology and concepts used in this thesis, we will give a brief overview of the terms used in the context of cybersecurity and TPR threat modeling.

Attack trees provide a hierarchical diagrammatic representation of attacks, starting from the primary goal of the attack down to the detailed steps and conditions necessary for the attack to be successful [65]. They make it possible to model attacks in various layers of detail, offering both high-level overviews and in-depth analyses. This granularity facilitates a better assessment of security risks and the development of targeted mitigation strategies. Attack trees can aid in prioritizing security improvements by allowing stakeholders to visualize and understand the potential impact of different attack scenarios, leading to more informed decision-making in cybersecurity defense planning [66]. Lallie *et al.* states that attack trees for visual representation in cybersecurity analysis offers a structured and intuitive method to identify and assess potential attack vectors within a system [67]. They are an effective tool for visualizing the paths an attacker might take to exploit vulnerabilities and help in comprehensively understanding the complex relationships between different components of a cyber system [67].

Attack vector, is a specific pathway or method utilized by adversaries to access a system, aiming to exploit vulnerabilities and encompassing the tools, actions, and human or technological susceptibilities within the target's environment, contributing to the broader attack surface [68]. Attack vectors enable attackers to carry out malicious activities, such as data breaches, unauthorized system access, or the delivery of malware [69]. The identification of potential attack vectors is a crucial step in threat modeling, as it helps in understanding how attackers might compromise a system and what measures can be taken to mitigate such risks [69]. Systematic mapping of attack vectors can offer insights into how adversaries operate and what measures can be taken to counteract these threats effectively within enterprise systems [70].

Attack surface is the collection of points at the boundaries of a system, system element, or environment that could be exploited by an attacker to enter, influence, or extract data from said system or environment [71]. The concept of an attack surface in the context of computer security refers to the total number of points where an unauthorized user (*adversary*) can try to enter data to or extract data from an environment [72]. More formally, the attack surface of a software environment is the sum of the different points (attack vectors) where an unauthorized user can try to enter or extract data [72]. Some researchers have used Model-Based threat modeling systems to understand how different attack vectors described by CAPEC patterns could be structured to document threats systematically, guiding the selection of security measures to protect CPS [73]. However, there still exists a large gap in the understanding of cyberattacks on CPS and the research is ongoing [74].

Attack path delineates a potential sequence of exploitations an attacker might employ within a system, detailed graphically, often using cloud security data to highlight origins, destinations, and the criticality of the threat [75]. Attack path assessment is a form of risk assessment, which normally uses graph-based algorithms to generate visual pathways in node-like tree format to expose exploitable paths in a system [76–78]. For example, threat modeling method based on Threat Modeling Attack Path Analysis (T-MAP), which quantifies security threats by calculating the total severity weights of relevant attack paths for Commercial Off The Shelf (COTS) systems, proves to be a valuable way of quantifying and reasoning difficult decisions [79]. Graph Views of Attack Paths created by Chen *et al.* allow easy visualization of the attack paths and their associated business values [79]. The findings of Chen *et al.* suggest that not all attack paths are equal; some may be easier for attackers to exploit, and some may have more severe business value impacts [79].

2.4 Threat modeling and attack patterns

Formal threat modelling methodologies and threat classification models such as CLASP, SDL, STRIDE, DREAD, and TAM [80] which can be used to describe cyber threats are often over simplified to a generalized qualitative value range [74]. Yet, formal models provide a precise, machine-readable way to represent the system and its potential vulnerabilities, enabling automated and systematic threat analysis [80]. Informal methods, while less precise, are still valuable for brainstorming, risk assessment, and communicating complex ideas in an understandable manner [80, 81]. The complexity of any system itself presents difficulty to model the system as a whole [82] in addition to the problem of formulating effective defensive strategies or attack detection methods [74] which may result in our failure to detect cyber threats which have not been considered before. Thus, to get the most out of threat modeling, a combination of formal and informal methods could compliment each other [83, p. 275], as it proved valuable for Xu and Nygard when researching Petri nets [83].

Review of various modeling techniques by Lallie *et al.* warns that graphical modeling techniques also carry dangers of inefficient modeling in cybersecurity visual representations [67]. The failure of generating un-standardized, visually complex and misleading graphs, can compromise the understanding and analysis of cyber threats [67, p. 30]. A neutral, node-based structure should facilitate easier understanding of the flow of the attack paths and reduce cognitive load for the reader.

2.5 CAPEC, CWE and CVE building blocks

MITRE hosts and maintains the CVE [84], CWE [85], and CAPEC [86] databases to provide standardized references for identifying, describing, and classifying security vulnerabilities and attack patterns. This ecosystem allows for the effective communication and management of security information among various stakeholders in the cybersecurity community.

Table 3. Catalogues information.

Catalog Name	Version	Set size	Data size	Data type
CAPEC	3.9	615	3,21 MB	JSON
CWE	4.13	959	13,5 MB	JSON
CVE	25.02.2024	304765	346 MB	JSON

Set size reflects total size of unique items in the catalog.

CAPEC provides a comprehensive framework for identifying and understanding a wide

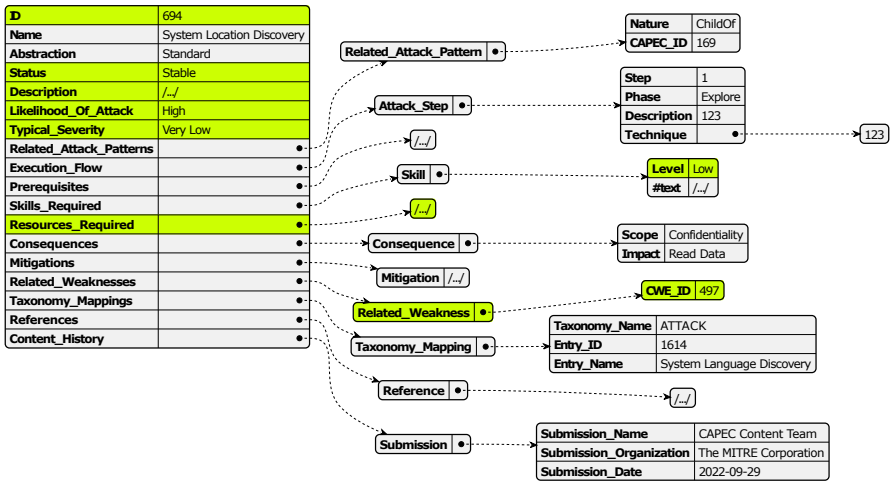


Figure 6. Example of a pattern item in JSON format.

array of attack patterns across different domains [86]. Threat modeling is crucial for bridging the gap between domain and security expertise, but often lacks guidance and formalization, making it complex for non-security experts [87]. In asset based threat modeling using CAPEC, the hierarchical structure of attack patterns can be a useful substitute for the traditional brainstorming sessions [87]. Example of a single item can be seen in Figure. 6. But in addition host a catalogue with 615 individual patterns (see Figure 3), the catalogue also provides a hierarchical structure, or categories, to said attack patterns. CAPEC items are categorized into domains such as Software, Hardware, Communications, Supply Chain, Social Engineering, and Physical Security, which represent the primary areas of vulnerability that attackers may exploit. CAPEC domains encompass attack patterns at varying levels of abstraction, ranging from meta patterns that outline broad methodologies to detailed patterns focusing on specific techniques. For example, some studies have used CAPEC to identify attack patterns in IoT devices [41], while others have used it to analyze the security of web applications [87]. This analytical approach, facilitated by CAPEC's structured enumeration of attack patterns, enables a comprehensive understanding of potential attack vectors in systems.

CWE is a categorization of software and hardware weakness types [85]. It provides a common language for describing security vulnerabilities in code, aiming to help developers prevent such vulnerabilities, by linking to CAPEC or CVE entries (see Figure 7). Studies which explore the practical relationships between CWE and system's vulnerabilities focus on formal threat modeling methods(STRIDE) [88], use CVSS like metrics [89], or other hybrid methods that combine formal and informal threat modeling techniques [90], to create associations. CWE main purpose intended to facilitate communication [91], it alone is not enough to provide a comprehensive understanding of the attack patterns and vulnerabilities in a system, as it fails to provide the coverage and owner-viewpoint [92]. It

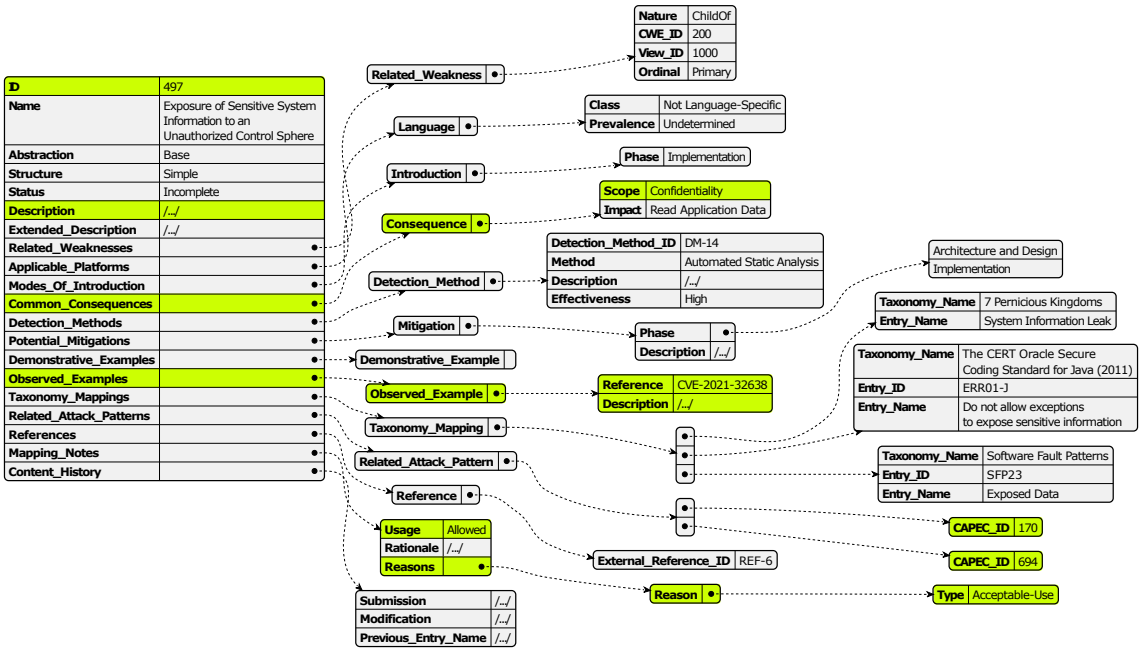


Figure 7. Example of a weakness item in JSON format.

can however, be used as a comprehensive knowledge base for identifying and mitigating the most dangerous types of vulnerabilities in software [93].

CVE is a list that provides unique identifiers for publicly known cybersecurity vulnerabilities [84]. Each CVE entry contains an identification number, a description, and at least one public reference for publicly known cybersecurity vulnerabilities (see Figure 8). The main goal of CVE is to make it easier to share data across separate vulnerability capabilities (tools, databases, and services) as part of an international effort to improve cybersecurity. CVE Entries prove to be invaluable, as they can help to manage cybersecurity risks [94], tracing relationships between CAPEC patterns using NLP techniques [95], or even in some cases, assist in predicting software vulnerabilities [96, p. 229].

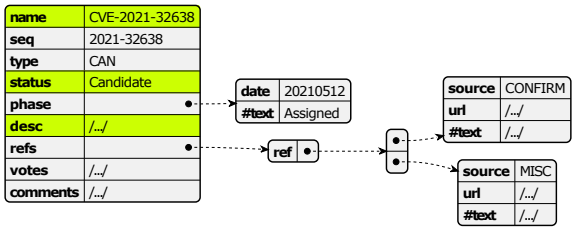


Figure 8. Example of a vulnerability item in JSON format.

By combining and mapping the CAPEC, CWE, and CVE catalogue items, researchers can gain understanding of the relationships between attack patterns, weaknesses, vulnerabilities, and real-world exploits.

2.6 Natural language generation and threat scenario generation

NLG capabilities in LLMs are harnessed as agents to automate computer tasks and address complex problems via natural language instructions, offering a more adaptable and scalable solution than conventional approaches that necessitate extensive expert demonstrations and task-specific reward functions [97]. Generative AI is likely to have significant impact cybersecurity and privacy, as conversational abilities of ChatGPT can revolutionize how we interact with technology [98]. When applying NLG to the security requirements domain, researchers evaluation through a user study demonstrated a 100% task completion rate, with 57% full accuracy and 43% partial accuracy, underscoring its potential utility for diverse technological assessments [99]. NLG has also proved useful in automating cyber threat intelligence reports [100]. Dual aspects of offensive and defensive strategies in Generative AI's application to cybersecurity, highlight the transformative potential to enhance cybersecurity measures while also cautioning against the proliferation of advanced tools for malicious use [101].

Automated tasks like NLG can also be susceptible to other adversarial effects too, such as hallucinations [102], social bias [103] and other open research problems [104]. Hallucinations refer to LLMs fabricating non-existent facts or generating inappropriate responses, which undermines trust in their output [102]. In the study by Yao *et al.*, showed that even nonsensical prompts composed of random tokens can induce LLMs to hallucinate [102]. This can present a problem when prompting LLMs to generate threat scenarios and supplying models with unstructured(threat modeling) data. Kim *et al.* has said that hallucinations is attributed to the fundamental limitation that LLMs cannot learn all computable functions accurately, leading to inevitable discrepancies between the model's outputs and the actual truth [105]. Regardless, some researchers find LLMs to be useful in cybersecurity as LLM-based threat modeling system was able to create responses which met human evaluators' expectations over 75% cases [106, p. 2]. While the ability to generate reliable text is improving for LLMs, problems towards trust and authenticity issues are on the rise, as it's becoming more difficult to distinguish between human and machine generated text [107].

2.7 Sentiment analysis and semantic similarity

Sentiment analysis, within the context of assessment methodologies [108], plays a pivotal role in determining the applicability and relevance of possible cybersecurity issues [109, 110]. This approach goes beyond traditional binary or numerical assessments, incorporating an evaluative layer that interprets the sentiment or contextual suitability of natural

human language [111]. Sentiment analysis can aid in cybersecurity decision-making in various ways. For example, to analyze cybersecurity aspects of Business Intelligence (BI) [112], to assess cybersecurity related content from social media [113], or to classify ambiguous network activity as threatening or innocuous [114]. Processing unstructured healthcare ecosystem security data using NLP can enable the identification, assessment, and management of emerging cyber threats [115]. To that end, researches have begun to explore the potential of sentiment analysis in cybersecurity with publicly available LLMs. ChatGPT [116–121], llama-2 [118–120] and gemini-1 [120], [121] pre-trained models show a promising level of performance and reliability as a sentiment analysis tool - same models are used in given thesis in Chapter 3. Incorporating sentiment analysis into threat modeling process, can hence be a promising method to aiding in the selection cybersecurity decisions [116], [122], in the given thesis it was used to ensure that connections between CWE, CAPEC and TPR system component are relevant (see Subsection. 3.3.5).

Semantic similarity in computational linguistics, the advent of the Universal Sentence Encoder (USE) has precipitated a paradigm shift in the analysis of text-based similarity. As delineated in the work of Cer *et al.*, the encoder's proficiency in transfer learning allows for a substantial reduction in requisite training data, fostering an unprecedented flexibility for academic research [123]. The provision of both transformer and Deep Averaging Network (DAN) models within USE affords researchers the latitude to navigate the trade-off between computational efficiency and accuracy, tailoring the tool to diverse research exigencies [123]. The transformer variant of USE employs attention mechanisms to generate contextually enriched sentence embeddings, transcending the capabilities of traditional word-level embeddings in capturing semantic nuance [124].

The multitask learning framework underpinning USE augments its robustness and amplifies its generalizable across disparate datasets. This trait is particularly salient for academic endeavours characterized by data paucity or those probing the frontiers of linguistic phenomena [125]. The accessibility of USE via TensorFlow Hub [126], complemented by its open-source status, enhances its utility, inviting widespread adoption and community-driven evolution. USE formidable performance—eclipsing models devoid of transfer learning and those limited to word-level transfer—underscores its superiority in generating high-fidelity sentence embeddings [123]. Such embedding technique is widely regarded as the major break, as a vector can be translated into a relatively low-dimensional space value, known as an embedding which can be subsequently used to captures some of the semantics of the input [127, p. 1757]. The academic community's embrace of USE is thus a testament to its efficacy, versatility, and the promise it holds for advancing the analytical capabilities of text-based research within and beyond the confines of natural language processing.

3. Methodology

The methodology section of this thesis provides a detailed exploration of the processes used to model potential cyber threats, starting with the initial design and moving through various stages of data collection, analysis, and validation. Each phase is described by showcasing procedure on a small portion of data, highlighting the tools and techniques employed to identify and refine attack paths, thereby offering the reader a clear understanding of how attack paths are generated and evaluated. The following sections describe details of each step of the process, beginning with an overarching research design and continuing through the individual phases of the study. This structure ensures a thorough exploration of the attack paths from their conception to validation.

3.1 High-level overview of research design

Research design is divided into five phases, of which one phase builds on the previous one and the final presentation phase (see Chapter 4) is the visual demonstration of the output.

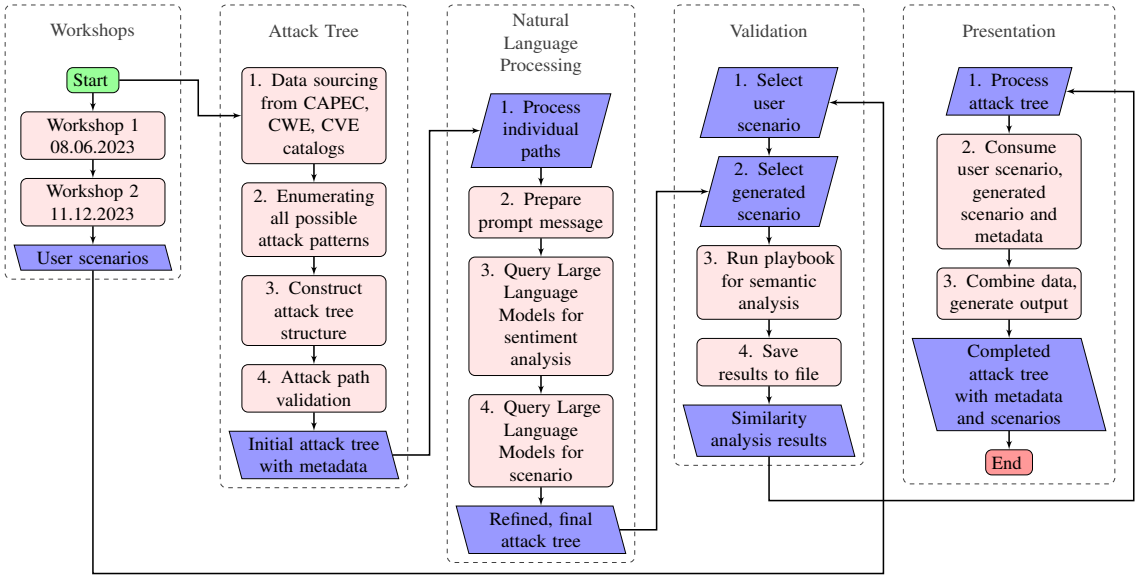


Figure 9. Research design flowchart.

Phase 1: threat modeling workshops: First phase begins with threat modeling workshops, utilizing the Elevation of Privilege (EoP) Threat Modeling Card Game to stimulate creative thinking in generating potential threat scenarios. The process started with interactive sessions where participants engage in threat modeling exercises using the EoP card game. In total 2 sessions were held with participants with medical and technical backgrounds who are familiar with TPR technology. Participants were tasked to fill out a questionnaire which

included sections with general information about the participant and their description of a possible attack scenario. The results gathered from these sessions are used in phase 4, where we validate the plausibility of the generated attack paths, by comparing how similar the generated attack paths are to the user-generated attack scenarios.

Phase 2: generating attack paths: Second phase starts with data sourcing for the modeling process. Data is sourced fom CAPEC, CWE and CVE catalogues. Identified relevant data points are extracted, normalized and are loaded into a local database for processing. Data processing is run by Java 18 Spring Boot application [128]. Algorithm 1 processes the data and generates potential attack paths by utilizing all available catalogues and the domain relations. The output of this phase is node-based graph with data points, which will give the hierarchical structure of the attack paths in an unprocessed format. The attack tree structure is further refined by additional validation steps to filter out paths that don't meet the requirements of the model.

Phase 3: natural language processing: Second algorithm uses the output from the first algorithm to construct prompt messages for NLP, transforming the technical attack paths into scenarios that are human readable. This step bridges the gap between raw modeling data and human interpretation, making the findings accessible to a wider audience and enables us to apply semantic similarity analysis in the next phase.

Phase 4: validation: The validation stage involves comparing NLP generated scenario descriptions against user-generated descriptions, employing another algorithm for SSA. This comparison helps in assessing the plausibility and validity of the generated attack paths with a probability score. The higher the probability score, the more likely it is that the generated attack path is plausible and the models output as a whole is trustworthy.

Phase 5: presentation of results Finally, the research ends with the presentation phase, where the generated attack paths are visualized in a node graph format. This visual representation, which includes data from all the previous stages, allows for a detailed analysis of the attack paths, highlighting key areas like potential choke points and high-value targets. The interpretation of the visual representation and the implications of the findings are discussed in conclusions section of the thesis.

This integrated approach, combining creative workshops, algorithmic processing, NLP, and validation techniques, offers a comprehensive and multi-faceted perspective on cyber threat assessment. The thesis thus aims to contribute to the field of cybersecurity by providing a novel methodology for identifying and analyzing potential cyber-physical threats.

3.2 Phase 1: threat modeling workshops

Main goal of the workshops was to gather data on potential attack scenarios from participants with medical and technical backgrounds who are familiar with TPR technology. User scenarios were used to validate the generated attack path's scenarios (see Figure 16) using Algorithm 3. The workshops were organized as a part of scenario testing [8] or conference. Topics at the conferences included the use TPR technology and it's applications, but more importantly the participants were allowed to familiarize themselves with the technology by interacting with the devices physically. Participant selection to the workshops was on a voluntary basis from the conference attendees, and they remained anonymous. In total 17 participants took part in the workshops. Out of whose 8 were male (average age of 32.5 years) and 9 were female (average age of 34.2 years). The recording of age and gender, which might raise ethical concerns, was primarily used to ensure a diverse demographic representation and to illustrate the participants were not minors. This demographic data was not analyzed further as the analysis of participant profile was not the main focus of the study.

Participants were given a questionnaire to fill out (see appendix 2 for an example), which included sections with general information about the participant and their description of a possible attack scenario. Usually, threat modeling is a process that is done by security professionals, as they have the domain knowledge to identify potential threats [129]. However, since the domain of TPR technology is relatively new, it is important to also involve people who are not security professionals in the threat modeling process, but still have a good understanding of the technology. In order to get the best quality data, the participants were given a brief introduction to the EoP card game and were given guidance on how to formulate structured threat scenario descriptions. EoP card game (see figure 10b) is designed to draw people who are not security practitioners into the craft of threat modeling [129]. Game was developed by Shostack in 2010 [129]. EoP card game consisted of 84 playing cards on topics about Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege which is based on *STRIDE* threat model [129]. Deviation from the standard EoP card game was that the cards were not distributed to the participants, but were drawn from a randomized pack by the participants. This card was used as a starting point for the participant to think about a possible attack scenario.

In order help participants to formulate a structured threat scenario descriptions, a small example was included on the questionnaire. Burger *et al.* proposed a five-layer taxonomy for classifying threat-sharing technologies with the classical description of the basic 5W's (who, what, when, where, and why) [130]. Structured example as see on figure 10a

"An attacker could exploit **[weakness]** in **[component]** to **[action]**, leading to **[consequence]**"

"An attacker could exploit insufficient authentication in the robot's remote access protocol to gain unauthorized access, potentially leading to data compromise."

(a) Structured threat scenario description example from the questionnaire.

(b) Elevation of Privilege card game cards used in threat modeling workshops.

Figure 10. Threat modeling workshop's elements.

is a simplified version of *STIX* based threat description [40], which is a standardized way to describe threat scenarios. The 5W's give the big picture: above the layer of the individual incident or intelligence, the 5W's correlate multiple incidents, some that may seem unrelated [130]. Together with 5W's and *STIX* Tactics, Techniques and Procedures (TTP) the example on figure 10a serves as a structured threat scenario description example. The combination of a structured threat description and the EoP card game should be sufficient to help even the most novice threat modeling participants to formulate plausible structured threat scenario descriptions.

Prior to using threat modeling workshop questionnaire (as seen on Appendix 2), initial questionnaire version was piloted on a small group on cybersecurity researchers / lecturers working at Tallinn University of Technology in 17.05.2023. From the feedback of the pilot it was evident that the initial questionnaire was too long and the questions were too complex. Thus, a new version of the questionnaire was created, which consisted of an example of what a structured threat scenario description could look like and a free section form for the participants to fill out. Pilot questionnaire results were not included in the study.

3.3 Phase 2: generating attack paths

The existence of a vulnerability in one component or part of a layer in the system does not mean that there exists a path for an attack. In order for an attack to be possible, the attacker must have a valid path to the target. Thus, we need to map TPR components in their respective layers and the interactions between them. The hardware components and I/O interfaces are the physical building blocks of the TPR (see Figure 4) and are the main

targets through which potential cyber(physical) attacks will be carried out in the study. By using all available information regarding access to components (see Figure 4), adversary capabilities (see Figure 5) and domain relations (see Figure 6) we will be able to construct all interactions (see Figure 11). The following section describes the modeling process by showcasing proposed modeling method applied on a small part of the whole dataset.

3.3.1 Mappings

Modeling focuses on the interactions between the components and elements which are required for potential attack paths. For that reason we assign a unique identifier (*ID*) to each element. For example, component *Cameras* (denoted as $\mathcal{C}$) can be represented as $\mathcal{C}_2$, or adversary *Insider* (denoted as $\mathcal{A}$) can be represented as $\mathcal{A}_2$. Other model elements (Domain $\mathcal{D}$, Target $\mathcal{T}$, Pattern $\mathcal{P}$) follow the same naming convention.

Components used for mapping originate from the fictional TPR device as seen on Figure 4, which is a representation of a typical telepresence robot's hardware components and I/O interfaces derived from common, real-world telepresence robot designs. We have introduced constraints that some components require physical access (USB Input Ports) or bluetooth access (Bluetooth module) to the component. This means that only adversaries, as seen on Table 5 with the capability to exploit these access types can carry successful attacks through these components. These constraints adhere to the real-world scenarios where physical or bluetooth access is required to exploit attack pattern based on the adversary's proximity to the device.

Table 4. Attackable TPR components (denoted as $\mathcal{C}$).

ID	Component	ID	Component
1	Vision Depth Modules	8	Microcontroller
2	Cameras	9	Multi-touch LCD Screen
3	Ultrasonic Range Finders	10	USB Input Ports **
4	Full Range Speaker	11	Drive Motors
5	Digital Microphones	12	Motorized Height Control
6	Wi-Fi Module	13	Deployable Kickstands
7	Bluetooth Module *	14	Lithium Ion Battery

* requires bluetooth access. ** requires physical access.

Adversary models are shown with different capabilities on Table 5. Naming convention is of little significance, but it is important to note that the capabilities of the adversaries are different which reflects the typical capabilities of real-world adversaries. Similar types of adversarial models in the health care ecosystem [41, p. 17][131, p. 6] are commonly used

in threat modeling related research to represent different types of attackers with varying skill levels and motivations. Healthcare ecosystems threat actors can be categorized into different groups based on their motivations, skills, and resources derived from official reports such as the European Union Agency for Cybersecurity (ENISA) Threat Landscape Report, Verizon DBIR and other sources. Adversary models and their capabilities used are based on the common descriptions of threat actors found in the literature and from mentioned reports. The ENISA 2023 Threat Landscape Report, with a reporting period from 2021-Q1 2023 (215 incidents) describes the threat actors in the health sector with the following percentages: Cybercriminals (60%), Unknown actors (26%), Insiders (non-malicious) (7%), Hacktivists (5%), and Insiders (2%) [35]. In addition, Verizon's 2023 DBIR generalizes and describes the threat actors as Organized Crime (73%), Other (18%), End Users (13%), State-affiliated (7%) out of 2489 reported breaches [34]. With descriptions from the reports, we have created a set of adversary models with different capabilities which reflect the common threat actors and their capabilities in the health sector. The proposed adversary model will be used in subsequent steps to determine if adversary has the capability to exploit a certain component or domain by adhering to the attributes (access type, resources, likelihood, skill) of the adversary.

Table 5. Proposed adversarial model map for healthcare ecosystem (denoted as $\mathcal{A}$).

ID	Adversaries	Access type	Resources	Likelihood	Skill
1	Activist	-	Low	Low	Low
2	Insider	B,P	Low	Moderate	Low
3	Script Kiddie	-	Moderate	Moderate	Low
4	Competitor	-	Moderate	Moderate	Moderate
5	Criminal	-	High	High	High
6	State	B,P	High	Low	High

B - has bluetooth access, P - has physical access.

1. Access type: If a component $\mathcal{C}$ requires physical access or Bluetooth access, but the adversary model $\mathcal{A}$ does not have the capability to exploit these access types, then the path is excluded. However, if the adversary model $\mathcal{A}$ has the capability to exploit the access types and the pattern $\mathcal{P}_{\text{Execution\_Flow}}$ step includes the keyword *physical*, or *bluetooth* then the path is considered valid.
2. Resources: If a pattern $\mathcal{P}$ requires resources ($\mathcal{P}_{\text{Resources\_Required}}$), but the adversary model $\mathcal{A}$ has resources of *moderate* (or lower), then the path is excluded.
3. Likelihood: If a pattern $\mathcal{P}$ has a likelihood of attack ($\mathcal{P}_{\text{Likelihood\_Of\_Attack}}$) of *high*, but the adversary model $\mathcal{A}$ has a likelihood of *moderate* (or lower), then the path is excluded.
4. Skill: If a pattern $\mathcal{P}_{\text{Skill\_Level}}$ (see Figure 6) requires skill level *high* but the adversary has skill level *moderate* (or lower), then the adversary $\mathcal{A}$ is not capable of exploit-

ing the pattern and such a path will result in invalid status and is excluded from subsequent steps.

The aforementioned adversary evaluation steps are executed by Algorithm 2 during the evaluation phase. This process is illustrated and explained in more detail in Subsection 3.3.3.

Domains in Table 6 in the context of HCS, are directly relevant to the system's primary targets: People, Software, and Hardware. By categorizing these patterns into specific domains and mapping them to HCS targets, when modeling domain relations, we can identify the potential impact of attacks on the HCS system's assets. The table mapping CAPEC attack domains to HCS targets illustrates a structured approach to identifying potential links between attack domains and system targets.

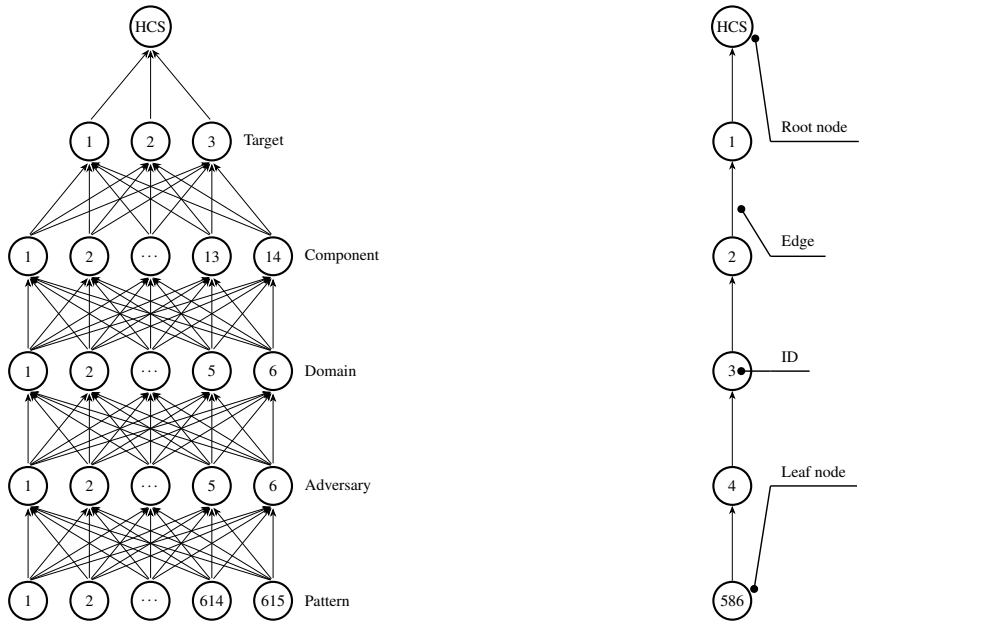
Table 6. Domains (denoted as $\mathcal{D}$) mapped to targets (denoted as $\mathcal{T}$).

ID	Attack Domain	ID	Target
1	Social Engineering	1	People
2	Supply Chain	2	Software
2	Supply Chain	3	Hardware
3	Communication	1	People
3	Communication	2	Software
3	Communication	3	Hardware
4	Software	2	Software
5	Physical Security	3	Hardware
6	Hardware	3	Hardware

For instance, Social Engineering attacks primarily target People, exploiting human factors to gain unauthorized access or information. Similarly, the Supply Chain domain impacts both Software and Hardware, reflecting the interconnected nature of modern systems where components and software sourced from various suppliers can introduce vulnerabilities. Communication attacks, encompassing a wide range of tactics from intercepting data to disrupting network services, pose threats to People, Software, and Hardware alike, demonstrating the pervasive risk of cyberattacks across all facets of a system. By analyzing realistic attack strategies which take into account the physical and cyber aspects of the system, interactions between system assets and the adversary capabilities, we can identify possible attack paths which could harm the system.

3.3.2 Generating paths

Using the previously mapped associations, we can now generate all possible attack paths by using data from CAPEC, CWE and CVE catalogues along with the mapped relations (see Figure 11). Initialization begins by enumerating all possible $\mathcal{T}$, $\mathcal{D}$, $\mathcal{A}$, $\mathcal{C}$ and $\mathcal{P}$ sets. Using Algorithm 1 to generate output, the result is a proper subset of $\mathcal{AP} = 929880$ different combinations which are all considered valid in this step. Selecting individual paths from set can be seen in Figure 11. For example, a single attack path ($\mathcal{AP}$), with $\mathcal{P}_{586}$, $\mathcal{A}_4$, $\mathcal{D}_3$, $\mathcal{C}_2$, $\mathcal{T}_1$ can be represented as: $\mathcal{AP}[586, 4, 3, 2, 1]$.



(a) Initialization of all paths.

(b) A single path ($\mathcal{AP}[586, 4, 3, 2, 1]$).

Figure 11. Step 1: Inizialisation of all paths.

3.3.3 Evaluation of paths

After generating all possible attack paths, we need to evaluate them to determine if they are applicable to the context. Using algorithm 2 we can determine if the generated attack paths are valid. Evaluation of 929880 paths is visualized in figure 12. Each path is individually evaluated by checking if the domain, target, CWE, CVE and adversary are valid. At the end of this stage, only valid paths(n=1580) are saved and used in the next phase. The reduction size from previous set is $\sim 99.83\%$. The following section describes the evaluation process.

Domain validation begins with iterating over CAPEC categories within the catalog. Each category represents a group of attack patterns that share common characteristics or ob-

jectives, serving as a high-level classification that aids in organizing and understanding the multitude of potential attack vectors. For each category, we check for two conditions: if the category contains relationships (indicative of associated attack patterns) and if the category status is not marked as *Deprecated*. Categories without relationships or those marked as *Deprecated* are bypassed, as they are either not relevant to current assessments or no longer considered valid within the catalog.

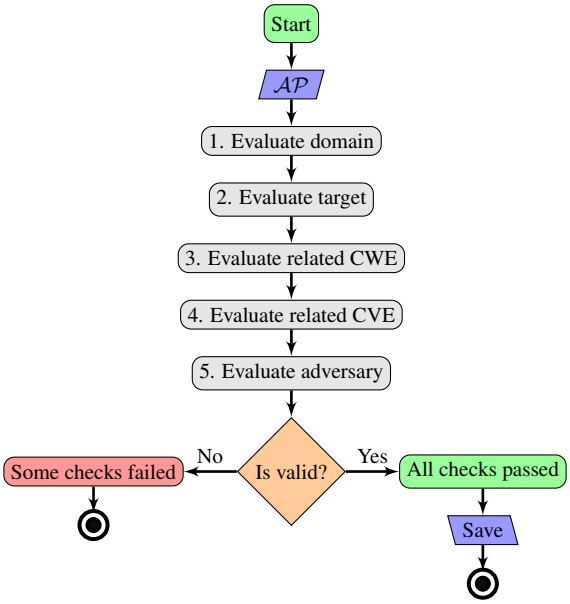


Figure 12. Step 2: Evaluation of paths.

Subsequently, we determine whether the specific attack pattern under evaluation is a member of the current category. This is achieved by checking the category's relationships for the presence of the attack pattern's ID. *Membership* in a category suggests that the attack pattern shares the common characteristics or objectives defined by that category. Upon establishing membership, the next critical step involves validating whether the category is associated with the domain under consideration. This validation relies on domain-to-category mapping logic, which correlates specific domains (e.g., software, hardware) with their relevant categories of attack patterns. If the attack pattern's domain matches the category's name, this indicates a direct relevance of the attack pattern to the domain in question and the path is marked as valid.

Target evaluation process aimed at determining the alignment of attack paths with specified HCS targets. Targets, such as *Software*, *Hardware*, or *People* within the HCS are compared against domain of the attack path. This is achieved by iterating over the list of targets which checks for the presence of the path's target within the domain's target list. Target and domain mapping domain relation is checked from Table 6.

CWE evaluation begins by examination of each weakness listed in the weakness catalog.

The focus is on those weaknesses designated as *Allowed* for consideration, indicating their allowed for mapping by the catalog's maintainers. Relevance is found by examining the relationships between weaknesses and attack patterns, specifically looking for a match between the attack pattern in question and any listed related attack patterns. CAPEC key *Related_Weaknesses-CWE_ID* is compared against CWE key *ID*. If an CWE is found for current attack path, it is marked as valid. When a match is found, it signifies a direct link between the attack pattern and a specific weakness, suggesting that the attack could exploit this weakness. Paths without a related weakness are marked as invalid due to a lack of relevant weaknesses.

CVE evaluation aims to identify specific vulnerabilities that may be exploited by an attack path. This evaluation extends the analysis beyond theoretical weaknesses (CWE) to consider documented instances of vulnerabilities that have been observed in the wild. CVE assessment closely follows CWE analysis, but focuses on the concrete examples of vulnerabilities.

The process being by mapping previously identified weakness within the context of an attack path to a related CVE. For each weakness under consideration, the evaluation seeks to identify any observed examples of vulnerabilities, referred to as CVE items. These CVE items are documented instances of security vulnerabilities that have been formally recognized and cataloged within the global CVE system. This step helps us to identify of if there exists a documented vulnerability that aligns with the weakness in question.

Attack paths that successfully map to one or more CVE items are validated as having a concrete basis in documented vulnerabilities, enhancing the practical relevance of the security assessment. Conversely, paths lacking related CVE items are marked as invalid from a vulnerability exploitation standpoint, due to the absence of documented vulnerabilities that align with the path's characteristics. By bridging the gap between theoretical weaknesses (CWE) and documented vulnerabilities (CVE), this evaluation helps to ensure that the attack paths are grounded in real-world security concerns, rather than purely theoretical mapping relationships.

Adversary evaluation assesses whether the adversary possesses the required skills to carry out the attack. This step is fundamental because the complexity of attack techniques varies, and not all adversaries may have the technical knowledge or expertise to implement them. CAPEC key *Skills_Required* value is compared against adversary model value. If the adversary lacks the requisite skills, the assessment deems the attack path as invalid, recording the reason as a lack of necessary skills.

Following the evaluation of adversary skills, the access evaluation process examines the necessity of physical access to certain components for the attack. This step is critical in scenarios where physical presence is required to exploit a vulnerability. If the adversary cannot gain the needed access, the attack is considered impractical under the current circumstances, and the path is marked invalid due to access restrictions. We compare if current adversary model at hand had access to the component as defined by Table 4 and Table 5.

Resource availability is another crucial factor in the adversary evaluation process. Successful execution of an attack often requires specific resources, such as hardware, software, or information. A lack of these resources renders the adversary incapable of carrying out the attack, leading to the path being invalidated due to insufficient resources. CAPEC key *Resources_Required* number of items is compared against adversary model.

Lastly, the likelihood of the attack's occurrence is assessed. This step considers various factors, including the adversary's motivation and the feasibility of the attack under existing conditions. An attack deemed unlikely to happen is marked as invalid, with the reason cited as its low probability. CAPEC key *Likelihood_Of_Attack* value is compared against adversary model value.

3.3.4 Normalization and merging of paths

Most catalogue values are represented in ordinal values, for example, CAPEC items have severity expressed as *very low*... *very high*. We assign a numerical value to each ordinal severity level from 1...5 and normalize the severity values using min-max normalization R_{norm} (see Equation. 3.1). $\mathcal{P}_{severity}$ (see Figure 13a) originates from CAPEC item key *Typical_Severity*(see Fig 6) value which was normalized.

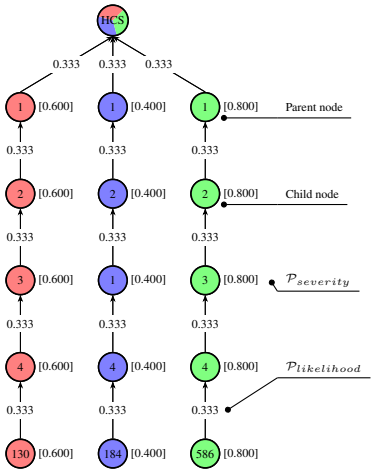
$$R_{\text{norm}} = \frac{R - \min(R)}{\max(R) - \min(R)} \tag{3.1}$$

Normalization allows us to compare severity ordinal values on the same scale (0...1) as other similar type of ordinal values (likelihood, skill). $\mathcal{P}_{severity}$ is normalized within the set by using R_{norm}. Similarly, $\mathcal{P}_{likelihood}$ is normalized and weight is equally distributed between all valid paths (each path is of equal weight at this stage). Merging nodes within an attack tree, as depicted in the Figure 13, is beneficial because it simplifies the complexity of the attack paths by consolidating information. This consolidation allows for a clearer overview of how different attack paths are interconnected and helps identify commonalities and variations among them while keeping the paths and integrity of the data. Merging of

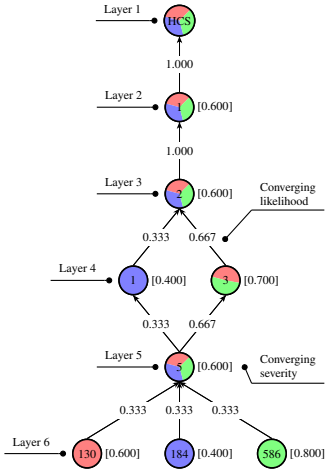
nodes is done by collapsing the nodes on layer basis by *ID* (see Figure 13b).

$$\overline{R}_{\text{avg}} = \frac{1}{n} \sum_{i=1}^{n} R_{\text{norm},i} \tag{3.2}$$

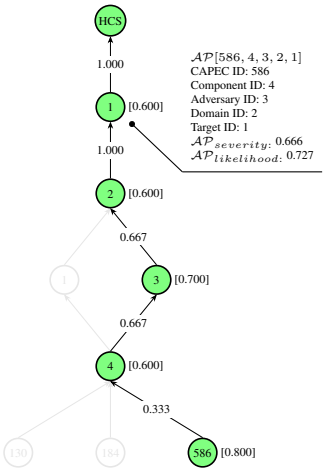
After normalizing and merging of individual nodes, we can calculate the average severity and likelihood for each path using $\overline{R}_{avg}$ (see Equation. 3.2).



(a) Example of a small section (3 paths selected).



(b) Step 3: Example of how nodes are merged for the same set of attack paths from Figure 13a.



(c) Step 4: Assesment of single path $\mathcal{AP}[586, 4, 3, 2, 1]$ selected from merged nodes.

Figure 13. Step 3: Merging nodes witin attack tree.

In addition to merging nodes, we also merge the values of the nodes, which will now in fact make it more clear how the paths are connected. When edges are merged by parent node *ID* and $\mathcal{P}_{likelihood}$ is merged by taking the average values from converging edges in the same path. The merging process and second normalization, for example, results in a path $\mathcal{AP}[586, 4, 3, 2, 1]$ where $\mathcal{AP}_{severity} = 0.666$ and $\mathcal{AP}_{likelihood} = 0.733$ are by merged edges (see Figure 13c).

This method is particularly useful in multi-criteria decision-making scenarios where various attributes need to be compared and weighted equally. After normalization, the average function helps in understanding the overall trend or central measure of the dataset that has been normalized, which is essential in further analysis like computing the overall risk or threat level by averaging the normalized severities, likelihoods, or skill requirements across all valid paths.

3.3.5 Sentiment analysis

This step is 1 out of 2 LLM prompts. The process involves analyzing the sentiment of textual descriptions related to attack paths metadata found on $\mathcal{AP}$ ($\mathcal{P}$, $\mathcal{A}$, $\mathcal{D}$, $\mathcal{C}$, $\mathcal{T}$). Sentiment analysis here does not adhere to its conventional use of discerning positive or negative emotions within text but is adapted to assess the fit or congruence between described attack patterns and potential targets. A positive sentiment in this context implies a high relevance or a strong likelihood that the attack pattern can exploit the associated weakness in the target component.

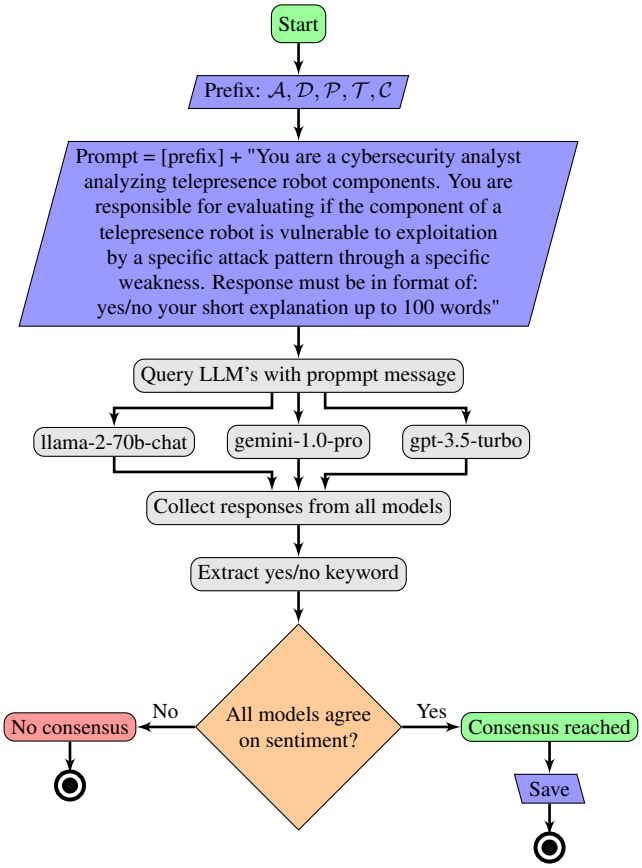


Figure 14. Step 5: Sentiment analysis for consensus.

Sentiment analysis is applied as a filter to narrow down the array of possible attack paths to those most pertinent to the assessment's focus. This is achieved by leveraging LLMs to interpret the context and implications of each attack pattern in relation to documented weaknesses and vulnerabilities. The sentiment analysis step is quantitatively measured, where paths validated through positive sentiments are deemed relevant, significantly reducing the number of paths under consideration.

Example of this process can be seen on Figure 14. This involves calculating the proportion of positive sentiments form each model, and determining consensus as positive when

all models agree on the sentiment. From the textual responses, the sentiment analysis process extracts the sentiment of each response by finding keywords (*yes* or *no*) from the start of sentence. Results are categorized as positive (*true*) or negative (*false*) and saved to local database. Such an analysis provides a granular understanding of the consensus among different evaluative models regarding the applicability of attack patterns to certain weaknesses or components.

The methodology extends to the aggregation of scenarios where all evaluated models concur on the positive sentiment, indicating a strong alignment or agreement on the relevance of attack patterns. This consensus approach strengthens the confidence in the already valid attack paths, ensuring that the final analysis is based on a robust evaluation of potential threats.

The incorporation of sentiment analysis into security assessments allows for a nuanced and context-aware selection of attack paths. By focusing on the compatibility of attack patterns with specific weaknesses and vulnerabilities, the process ensures that security resources are allocated efficiently, targeting the most significant and relevant threats to the system or component under review. After this stage 312 out of 1580 paths are valid. This results in an additional reduction of valid paths by $\sim 80.25\%$.

3.4 Phase 3: threat scenario generation using NLP

This step is 2 out of 2 LLM prompts. This phase involves feeding generated $\mathcal{AP}$ metadata ($\mathcal{P}$, $\mathcal{A}$, $\mathcal{D}$, $\mathcal{C}$, $\mathcal{T}$) into three separate language models (see Figure 15).

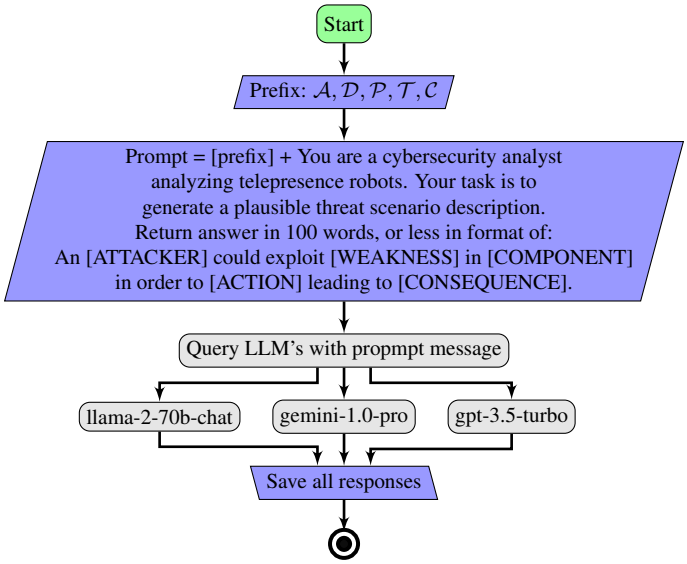


Figure 15. Step 6: Computer scenario generation.

The goal is to translate complex, yet structured data into narrative formats that are intuitively

understandable for the reader. Iterating over all valid paths (n=312), we construct a prompt message for each path. Instruction accompanies the attack path metadata, guiding the models on the desired structure and tone of the response. This ensures that the output aligns with the objectives of readability and comprehensibility, adhering to a predefined format that maintains consistency across scenarios.

Upon generation, the responses from each model are stored in a local database. This storage facilitates easy retrieval and comparison of scenarios, serving as a repository for the next phase of analysis. It also ensures that the generated content is preserved for further processing or review.

The stored scenarios are subsequently used in an analysis to find semantic similarities. This phase aims to identify overlaps or thematic consistencies across the narratives generated by different models, highlighting common threads or divergences in the interpretation of the attack path metadata.

3.5 Phase 4: text based semantic similarity analysis

The similarity analysis aims to evaluate the semantic similarity between sets of sentences across two datasets using NLP techniques. This analysis is driven by the objective to match responses from a primary dataset, containing unique user scenarios against computer generated scenarios in a secondary dataset.

The execution of the semantic similarity analysis is structured within a Jupyter Notebook, denoted as *playbook.ipynb* [132]. This digital environment facilitates an interactive computing workflow, where Python serves as the foundational programming language for the execution of the analysis. The execution leverages a suite of Python libraries, each contributing a unique functionality to the analytical process: TensorFlow and TensorFlow Hub: The core of the execution relies on TensorFlow, an open-source platform for machine learning, and TensorFlow Hub, a repository for machine learning models. Together, they enable the loading of the Universal Sentence Encoder (USE) model [123]. Pandas: This library offers extensive capabilities for data manipulation and analysis. Scikit-learn: Known for its robust machine learning algorithms, Scikit-learn is utilized to compute cosine similarity between sentence embeddings [133]. NumPy: A fundamental package for numerical computation in Python, NumPy supports high-level mathematical functions. The process begins with the text embedding phase, where sentences from both datasets are converted into numerical vectors (see Algorithm 3). This conversion is essential for translating the textual data into a format amenable to computational analysis, enabling the subsequent measurement of semantic similarity. USE is employed for this purpose,

leveraging its capability to encode text into high-dimensional vectors that capture the contextual and semantic nuances of sentences.

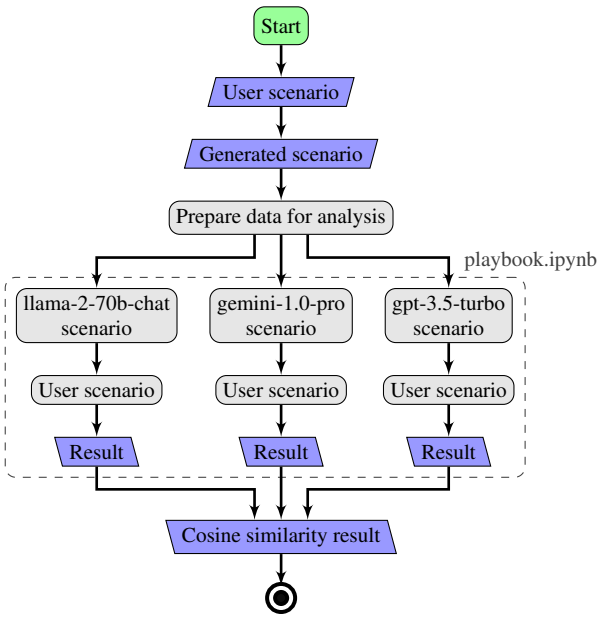


Figure 16. Step 7: Semantic similarity analysis.

Following the embedding, the analysis proceeds to calculate the semantic similarity between the embedded vectors, utilizing cosine similarity as the metric. Cosine similarity measures the cosine of the angle between two vectors, providing a real-valued score that quantifies their semantic closeness on a scale from 0 (indicating no similarity) to 1 (indicating identical semantics). This metric is chosen for its effectiveness in capturing the likeness in the meaning of text data, despite variations in phrasing or structure.

Outputs of sentence pairs is ordered by descending similarity scores. For later use, output values are linked with source IDs and similarity scores are saved to a local database. This list is intended to highlight the most semantically congruent pairs between the two datasets, facilitating insights into the relationships between individual responses and predefined scenarios. The results are compiled into a CSV file [134], which includes columns for both the source data and the computed similarity scores, providing a comprehensive overview of the analysis outcomes.

4. Results

During the modeling process we initially generated 929880 possible attack paths (see Figure 17), a combination of 6 adversary models, 615 attack patterns, 6 domains, 14 components and 3 targets. Initial paths were validated though several checks and needed a positive result from all checks for final consideration. After formal validation 1580 paths remained. These results were further refined by informal validation - sentiment analysis where 3 different LLMs agreed on sentiment if the attack pattern is applicable with the weakness and component in the context. End result was 312 valid attack paths (see Figure 17). From the 312 valid attack we queried 3 different LLMs to generate a realistic attack scenario. We would have expected 936 different scenarios, however, only 934 were generated. Model gemini-1.0-pro was unable to generate 2 scenarios from given prompt. Reason was, that safety guardrails were triggered and the model was unable to return an answer due to *harmful content*. The statistical impact of missing 2 scenarios out of 936 is negligible, this anomaly was not further investigated. We collected 17 unique user scenarios from threat modeling workshops and compared them against the generated scenarios(n=934) using SSA which internally used USE to calculate the similarity score.

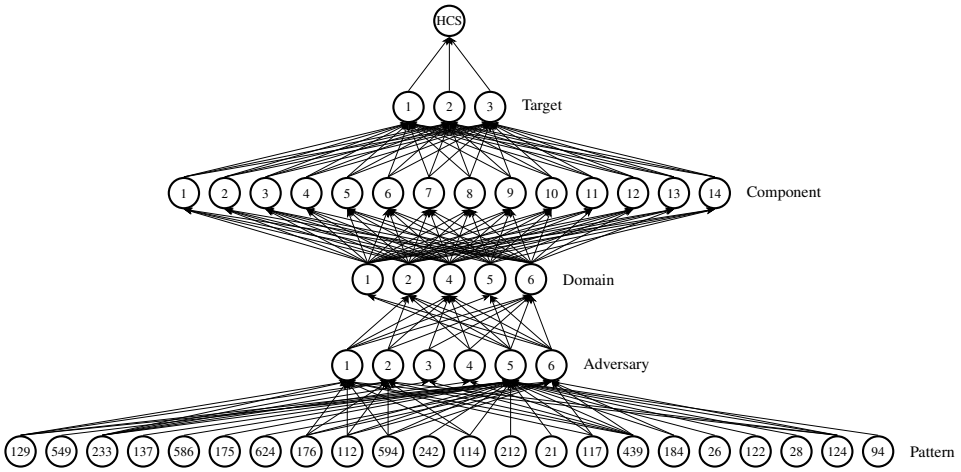
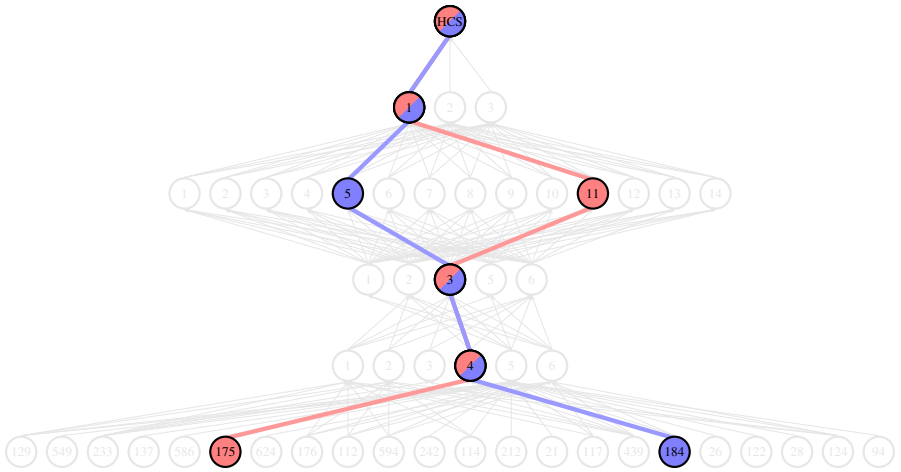


Figure 17. Results: Generated attack tree with valid attack paths.

To better understand the context and weight of a path, additional metadata (likelihood, severity, generated scenario) will help us to assess the seriousness and plausibility of a path. For example, lets observe most and least severe and likely attack paths (see Figure 18). The metadata accompanying each potential attack path not only contextualizes the threat, but also can aid in decision-making processes regarding the prioritization of security measures by focusing on the most severe and likely attack paths. For an in-depth threat analysis, the attack paths can be layered and cross-referenced against historical incident



Path	Severity	Likelihood	Adversary	Domain	Target	Model
$[175, 4, 3, 11, 1]$	0.764	0.419	Criminal	Software	Software	gemini-1.0-pro
Computer generated scenario: An attacker could exploit a vulnerability in the ultrasonic range finders to force arbitrary code to be retrieved locally or from a remote location and executed. If the vulnerability is successfully exploited, the attacker could cause the telepresence robots to execute arbitrary code, which could allow them to take control of the robots or access sensitive information.						

Path	Severity	Likelihood	Adversary	Domain	Target	Model
$[184, 4, 3, 5, 1]$	0.655	0.469	Criminal	Software	Software	gpt-3.5-turbo
Computer generated scenario: An attacker could exploit the weakness in the microcontroller's acceptance of path input with trailing slashes to manipulate the file system on a telepresence robot's software. This could lead to the attacker traversing to unintended locations or accessing sensitive files, compromising the integrity and security of the robot's software code and data.						

Figure 18. Results: Most severe and likely attack path (showing 2 out of 312).

data, providing a dynamic risk assessment model that adapts to evolving threats. The generated scenarios can be used to explain the attack paths in a more human readable manner, making it easier for stakeholders to understand the potential risks and implications of a given threat (all 312 individual paths are available at gitlab.cs.ttu.ee [128]).

The following sections describe observed results in a more holistic manner, providing insights into the generated attack paths and their implications.

4.1 User generated scenarios

Within two threat modeling workshops, participants were tasked with the creation of threat scenarios(n=17). The participant scenarios can be seen in Table 7. The author undertook the translation of 14 out of the 17 scenarios from Estonian to English, ensuring linguistic consistency and comprehension. These scenarios were subsequently utilized in a semantic text-based similarity analysis, juxtaposed against the entirety of generated scenarios (see Section 3.5).

Table 7. Results: User scenario answers.

ID	Gender	Age	Scenario
1	male	21	I use HTTPS, encrypt all messages. Actually, I have no clue.
2	male	24	An attacker can exploit repudiation to hinder developers productive time usage. If the developer does not know when the log entry occurred, they have no basis for error resolution. Missing data can also lead to errors.
3	male	23	An attacker can gain access to a user through a robot, which can cause harm by providing access to user data.
4	female	41	Hacking into the institutions network where the robot is located.
5	female	43	An attacker can exploit physical weaknesses of the device, rendering it incapacitated - disrupting Wi-Fi, cornering it, and blocking its movement. This requires human intervention, where the attacker can gain access to the control computer or accesses through a helper. The devices have limited vision, so it is not possible to see if the attacker is in the same room, for example.
6	female	30	An attacker can exploit inadequate data protection. The attacker can gain access to personal data, through which they can find personal information to create fake accounts on social media, loan accounts, etc. This results in a reputation loss through social media and also with banks.
7	female	38	For personal data usage, an attacker can steal personal data (codes, images, videos, etc.) to replace a person and control a remote participation robot.
8	male	23	Changing the mapped area or points on Temi robot, overloading or turning off Wi-Fi networks.
9	female	25	An attacker can exploit tampering with the engines pathfinding algorithm to make changes that can lead to the robot driving into a wall or a person at high speed.
10	male	35	An attacker could exploit generated call links to join the room/audience to retrieve sensitive information. If the Temi platform does not require hard authentication or normal acceptance of an incoming call, unwanted guests could initiate the call at an unwanted time.
11	male	35	DDOS type attacks: overcrowding the call platform with bot-calls to restrict/limit actually invited guests from joining.
12	male	35	Joining the call without a camera, no less option to authenticate the person if only a public link is shared.
13	female	35	Access (who has physical and virtual access), specific usage protocols, guidelines, Wi-Fi security. Electrical safety (batteries), collision sensors, speed limiting. Robot parking spots (to avoid hazards on the road), user rights.
14	male	64	Physically impacting the robot with objects, even a small bump can alter its behavior. Document and data theft. In both cases, access to the robot left unattended, accessible to a third party. Can be approved for malicious use.
15	female	32	The operator can record videos and engage in conversations with the robot. They can misuse it, for example, to sell information, demand money, or otherwise disclose it.
16	female	32	How to deal with operator credentials (e.g., passwords) for the robot? Restrict the operator from installing programs that record the screen, but they can still record with their phone.
17	female	32	How to ensure that the right person clicks on the links sent to their email? Why not develop authentication methods like using a phone, ID card, or smart ID for login.

Source files can be found at cs.gitlab.ttu.ee [135].

4.2 Attack tree

To generate all valid paths(n=312) total time was 377.6 minutes as seen on Figure 19. Out of which 255 milliseconds was used to initialize all combinations of attack paths and 27.9 seconds to validate them. Significantly longer steps included LLM prompts for sentiment analysis 314.6 minutes and to generate scenarios was 62.5 minutes.

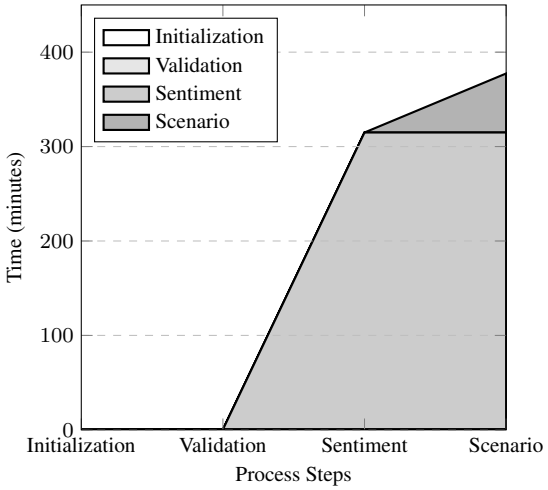


Figure 19. Results: Attack tree generation time.

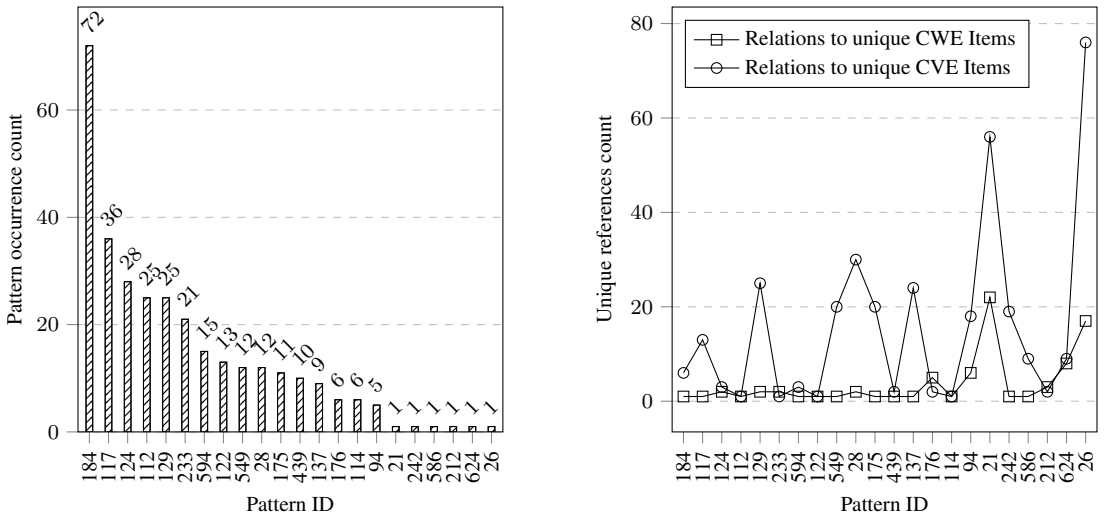
The following sections summarize the results of the attack tree by different views explaining the observed results and their implications.

4.2.1 Patterns results

In the analysis of attack paths, a total of 312 valid paths were identified, each originating from a distinct pattern at the first layer, serving as a common entry point for all attack paths. The merging process of attack paths by layers inherently led to the consolidation of the patterns, culminating in a final count of 22 unique CAPEC attack patterns, as depicted in Figure 20. This figure also illustrates the distribution and interrelations of these patterns with other catalog items. Among the attack patterns used, $\mathcal{P}_{184}$Software Integrity Attack(n=72) emerged as the predominant pattern, succeeded by $\mathcal{P}_{117}$Interception(n=36) and $\mathcal{P}_{124}$Shared Resource Manipulation(n=28). Conversely, patterns such as $\mathcal{P}_{21}$Exploitation of Trusted Identifiers, $\mathcal{P}_{242}$Code Injection, $\mathcal{P}_{586}$Object Injection, $\mathcal{P}_{212}$Functionality Misuse, $\mathcal{P}_{624}$Hardware Fault Injection, and $\mathcal{P}_{26}$Leveraging Race Conditions were identified with a solitary instance each.

Interestingly, patterns associated with CWE and CVE reveals that although $\mathcal{P}_{184}$ was the most prevalent starting point for attack paths, it was only associated with 1 weakness and

6 vulnerabilities. As opposed to $\mathcal{P}_{21}$, $\mathcal{P}_{242}$, $\mathcal{P}_{586}$, $\mathcal{P}_{212}$, $\mathcal{P}_{624}$, and $\mathcal{P}_{26}$, which were the least common entry point, yet was linked to the majority of the weaknesses and vulnerabilities. Though the assessment of weaknesses and vulnerabilities is not the primary focus of this study, these insights could provide a possible research directions for future works examining the underlying relationships between attack patterns and their associated weaknesses and vulnerabilities regarding TPRs.



(a) Pattern occurance.

(b) Relations to unique CWE and CVE items.

Figure 20. Results: Attack pattern statistics.

Given the nature of a TPR as a software-driven communication device with shared resources, the prevalence of most occurring attack patterns aligns with the intrinsic functionalities of the robot. Conversely, the less frequently observed attack patterns predominantly pertain to Injection-based attacks. Injection based techniques often serve as sub-techniques within broader context of tactics [136, p. 13]. For instance, executing an *Object Injection* necessitates preliminary interaction with the target system, enabling the injection of malicious content during the processing of serialized objects. This observation suggests that the robot's system architecture might not readily facilitate interactions capable of enabling such attack types, which in fact is a positive outcome.

4.2.2 Adversary results

Figure 21 depicts the distribution of adversaries across the 312 valid attack paths. The predominant adversary type identified was $\mathcal{A}_5$ (Criminal) 47.76%, followed by $\mathcal{A}_6$ (State) 25.32%, and $\mathcal{A}_1$ (Activist) 12.82%. Less frequent adversaries included $\mathcal{A}_2$ (Insider) 10.58%, $\mathcal{A}_4$ (Competitor) 1.92%, and $\mathcal{A}_3$ (Script Kiddie) 1.60%. The significant representation of state actor as potential adversary may initially seem unrealistic given the specific context of telepresence robots in healthcare when compared to a more modest number of

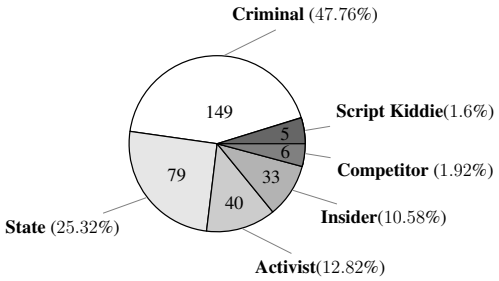
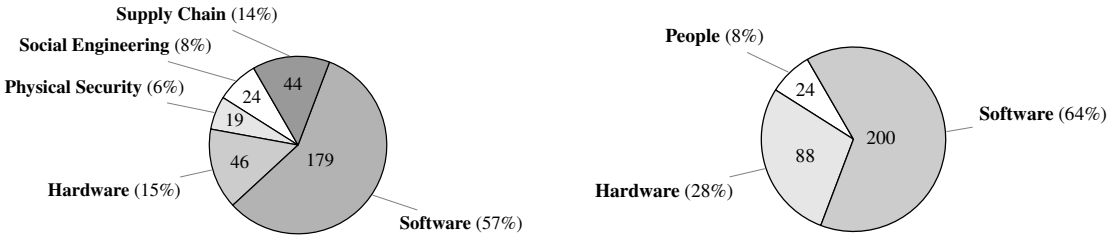


Figure 21. Results: Adversary occurrence count.

7% as reported by Verizon's 2023 DBIR [34] of nation state affiliated attacks. However, this result aligns with the theoretical capabilities of such adversaries, who are presumed to have substantial resources and advanced technological tools at their disposal, enabling them to exploit a wide range of weaknesses. While real-world attacks by state actors are not commonly reported, possibly due to their motives and utility leaning more towards covert operations [137], the threat modeling approach used in given thesis emphasizes the full scope of theoretical attack paths that could be exploited by highly capable actors. This modeling, therefore, serves more to highlight potential security gaps that could be exploited under conditions of high resource availability rather than to reflect the current trend in actual cyber-attack incidents. This discrepancy also somewhat suggests a possible over-representation in the input data used for modeling, which may skew towards more capable state-level adversaries. Therefore, we acknowledge, that this is the current model's limitation that it does not take into account the real-world context and likelihood of adversaries, but rather focuses on the theoretical capabilities of different adversary types.

4.2.3 Domain and target results



(a) CAPEC domain occurance.

(b) Number of paths reaching target.

Figure 22. Results: Domain and target attacks.

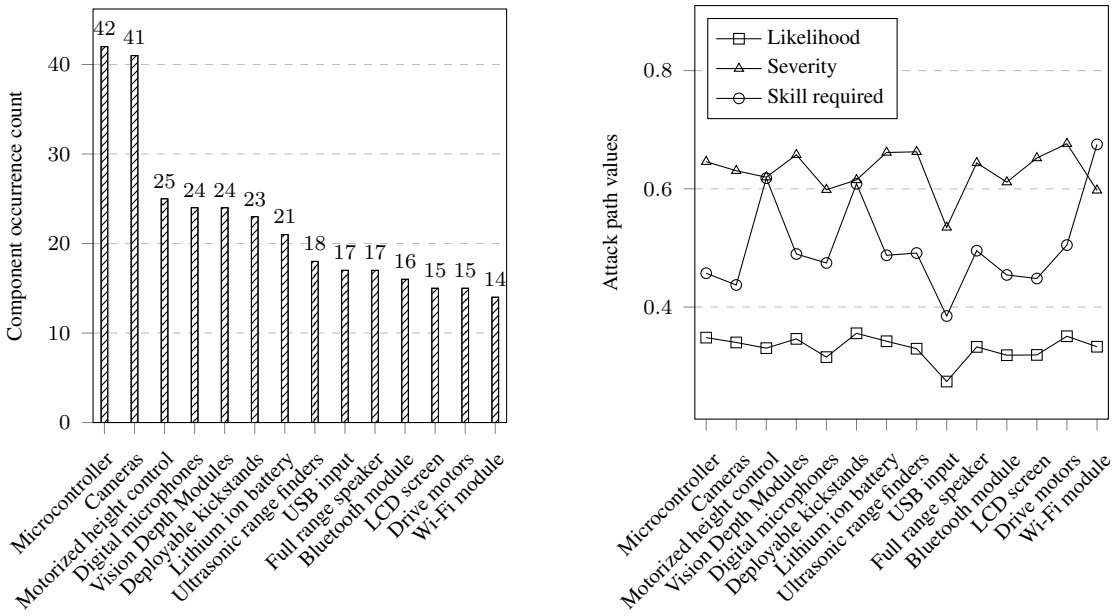
6 CAPEC attack pattern domain types were mapped to 3 target types (see Figure 6. So it is expected that the targets distribution closely resembles the domain distribution(see Figure 22). It will not, however, be a perfect match as some domains are more likely to be used in certain targets. When looking at the CAPEC domain types, the most used domain was the Software domain(57%). Similarly, attack paths which ended attacking target Software(64%) were the most common. This reflects the fact that the robot is a software

driven device and most of the attack patterns are software based. The least attacked target was People(8%) from domain Social Engineering(8%). This low number can be explained by the fact that although TPR can interact with people, it is not a primary target for social engineering attacks. Additionally, there doesn't exist a good relations between social engineering and field of robotics, as both are relatively new fields compared to software and hardware attacks, and the attack pattern analysis did not support such connections. Domain Communications is absent from the results completely, as it as all modeled paths involving the domain were invalidated for the same reason.

4.2.4 Component results

Out of generated 312 attack paths the most used components were microcontroller (n=42) followed by cameras (n=41) and motorized height control (n=25), which intuitively makes sense as these components are core elements of the robot. The least used components were LCD screen (n=15), drive motors (n=15) and Wi-Fi module(n=14). In the low end of list were also USB input (n=17) and bluetooth module (n=16). This is because those components require physical access to the robot for exploitation to be possible. As only 2 out of 6 adversaries have physical access and bluetooth access capability (see Table 5) that is reflected in the results. Surprisingly however, the least used component was the Wi-Fi module (n=14). We would have assumed Wi-Fi module to be more used as it is a common attack vector in IoT devices [138]. However, when looking at the average likelihood, severity, and skill requirement for each component (see Figure 23) we can see that the Wi-Fi module has the highest skill requirement(n=0.649) and one of the lower likelihoods(n=0.333) of it being used. That most likely explains the low number of times it was used in an attack as not many adversaries have the capability to exploit it. The lowest skill requirement (n=0.385) and severity (n=0.535) was for the USB input. Low skill requirement can be explained as users consider USB peripherals safe [139], they might inadvertently introduce a threat to the robot, easing the adversary's job. Interestingly the highest severity was for the drive motors (n=0.676). The drive motors are a critical component for the robot to function, and if they are compromised, the robot is rendered inoperable impacting the availability of the service. Or in the worst case scenario, the robot could be potentially used in alternative ways to cause harm to the environment or people (such as the cyber-physical risk showcased in Figure 5).

The data reveals a strong correlation between the likelihood of a component being exploited and the severity of the attack. This result stems from the fact that from the theoretical perspective, the likelihood of an attack is closely proportional to the adversary's capability to exploit a given component, increasing the severity of the attack. As severity is derived from CAPEC attack pattern, and likelihood is originating from the convergence of valid



(a) Component occurance, grouped by component.

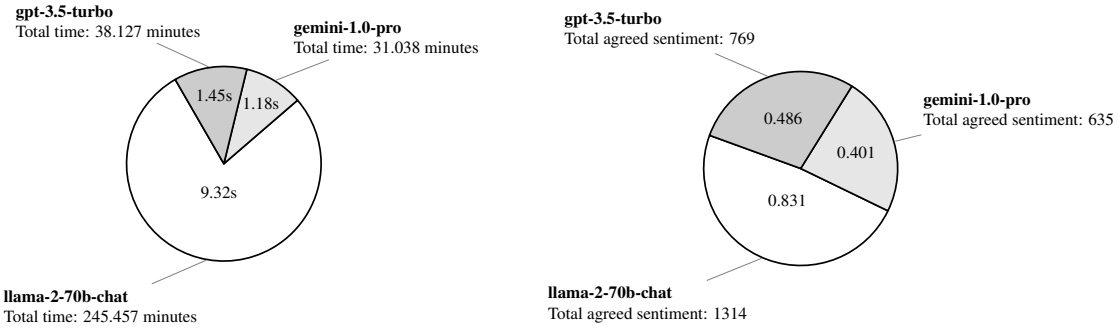
(b) Average likelihood, sererity, and skill requirement. Grouped by components.

Figure 23. Results: Component statistics.

attack paths (see Figure 13b) it is expected that the two metrics would be closely related. These results suggest that, proactive physical environmental analysis should be conducted, where the robot is likely to be deployed, addressing the potential risks which could arise from the manipulation of the robot's components which can interact with the environment. The choice of initializing the likelihood equally for all paths at initialization phase, and then merging likelihood based on the convergence of paths was to showcase the likelihoods of branches in the attack tree. For future research and threat modeling, likelihood of attack could also be potentially mapped from other catalogue items such as CAPEC *Likelihood of Exploit* or Common Weakness Scoring System (CWSS) *Likelihood of Exploit*. This could provide a more accurate likelihood of an attack pattern or common weakness being used by the adversary.

4.2.5 Sentiment analysis results

Sentiment analysis was performed on all generated paths(n=1580) with 3 different models after the formal validation, resulting in 4740 prompts (see Figure 24). Sentiment analysis was used to ask if the attack pattern is applicable with the weakness and component in the context(see Figure 14). Results were then used to filter out the unrealistic attack paths. The most agreeable model was the llama-2-70b-chat with 0.831% of the responses resulting in a true statement. More conservative models were gpt-3.5-turbo with 0.486%



(a) Average sentiment generation time in seconds.

(b) Average sentiment resulting in true statement.

Figure 24. Results: Sentiment analysis.

and gemini-1.0-pro with 0.401%. Llama-2 being more positive and gpt series being more neutral have also been observed in other studies [119, 120]. Only when all 3 models agreed on the sentiment, the attack path was considered valid, leaving us with 312 out of 1580 valid attack paths.

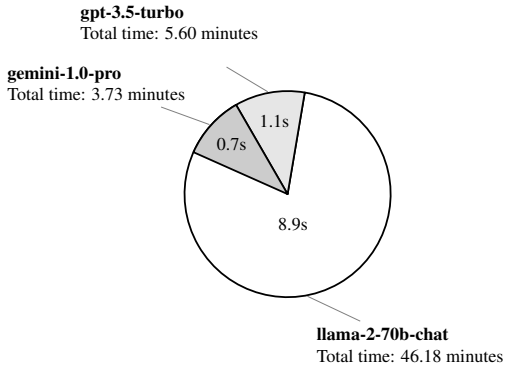
4.2.6 Generated scenarios results

Table 8. Results: Computer generated scenarios.

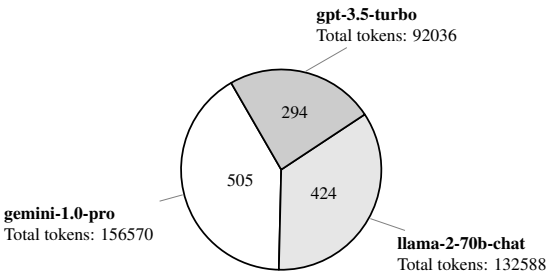
ID	PathID	Generated Scenario
1	[137,4,0,4,0]	An criminal could exploit weakness in the Bluetooth module by manipulating the content of request parameters to inject special characters, potentially adding or modifying parameters. This could lead to unauthorized access or manipulation of data on telepresence robots, compromising the privacy and security of users.
		...
934	[112,5,3,10,1]	An attacker could exploit the weakness of information sent over a network being compromised while in transit. They could use this weakness in the adjustable motorized height control component of a telepresence robot in order to gain unauthorized access to the robot's functionality. This could lead to the attacker being able to control the robot and potentially cause physical harm or compromise the security of the environment in which the robot is operating.

All scenarios available at gitlab.cs.ttu.ee [140].

LLMs gemini-1.0-pro, llama-2-70b-chat and gpt-3.5-turbo were prompted with metadata for NLP task to generate scenarios which at the end yielded in 934 unique scenarios. Expected output should have been 936 scenarios, however, 2 scenarios were not generated by the gemini-1.0-pro model due to safety guardrails being triggered by harmful content classification. The statistical impact of missing 2 scenarios out of 936 is negligible, this anomaly was not further investigated.



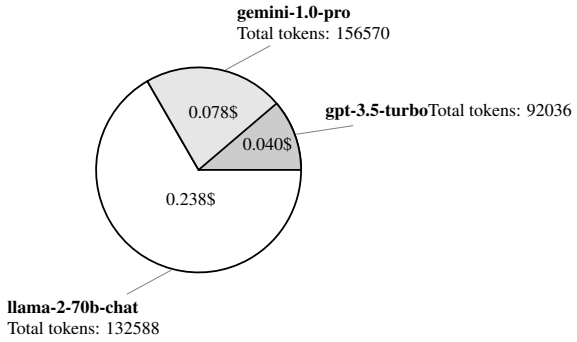
(a) Average scenario generation time.



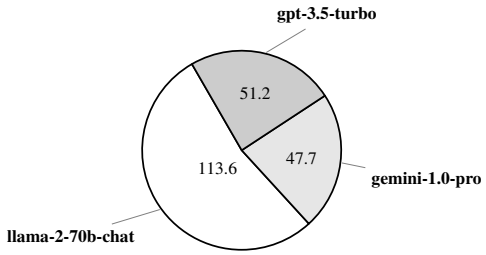
(b) Average token usage.

Figure 25. Results: Generated scenarios, time and tokens.

From a performance perspective, the gemini-1.0-pro model utilized a total of 156,570 tokens, the llama-2-70b-chat model used 132,588 tokens, and the gpt-3.5-turbo model consumed 92,036 tokens for responses. Respectfully, the response generation times for the models were 46.18 minutes, 3.73 minutes, and 5.60 minutes in total (measured from response sent till response received). These numbers, while not directly impactful to the results, are important to consider when selecting a service provider to carry out large scale scenario generation, where time-to-generate could be a critical factor.



(a) Total cost of generating scenarios.



(b) Generated scenario average word count.

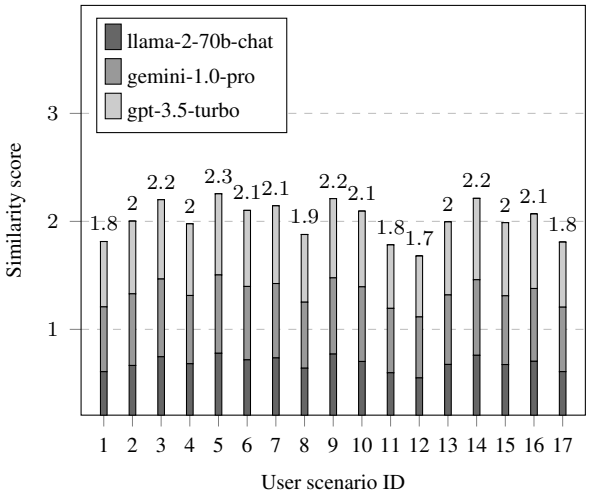
Figure 26. Results: Generated scenarios, cost and word count.

On average, the gemini-1.0-pro model's used tokens translated to a total cost of $0.078 [141], whereas the llama-2-70b-chat total expenditure was $0.239 [142]. The gpt-3.5-turbo model incurred a cost of $0.046 [143]. These figures illustrate the token consumption patterns and associated costs across different models, as seen on Figure 26, highlighting the economic considerations inherent in leveraging LLMs for scenario generation. Tradeoffs between cost and quality of generated scenarios should be considered when selecting a model for a given task, which was not the focus of this research, but is something to consider in future works. Interestingly, although all models received identical prompts with instructions to generate scenarios of up to 100 words, the token usage and resultant word counts varied across them. The average word counts per model were as

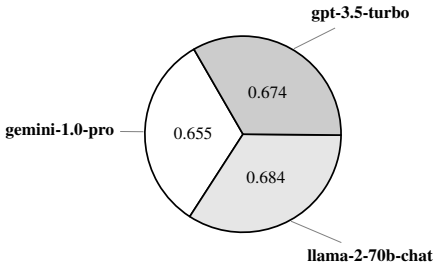
follows: gemini-1.0-pro at 47.7 words, gpt-3.5-turbo at 51.2 words, and llama-2-70b-chat significantly higher at 113.6 words, failing to follow instructions given in the prompt as seen on Figure 15.

4.2.7 Semantic similarity results

17 user scenarios were compared against all generated scenarios(n=312) with 3 different LLMs resulting in 15912 data points on semantic analysis. Lowest similarity score was 0.461 and highest 0.865. Fact that the lowest score is above 0.4 is a good indicator that the generated scenarios are not random. The overall average similarity score was 0.670 (see Figure 27b).



(a) Results: User scenario similarity scores.



(b) Results: Average similarity scores by models.

Figure 27. Results: Semantic textual similarity.

Relatively high average similarity score indicates that the generated scenarios are close to human generated scenarios. In fact, the average score was significantly higher than expected. These results confirm that the generated scenarios: 1) are relatively close to human-generated scenarios and 2) previous modeling steps were supportive to achieve this result. High semantic similarity score was achieved because the scenarios generated by the models were given detailed context metadata, and clear instructions, which helped the models to understand the assignment. The instructions set in the prompt as seen on Figure 14, followed similar structure which was given to the participants in the threat modeling workshops (see Figure 10a).

4.2.8 Limitations

1. Repudiation of the study is considered as limitation. The use of 3rd party LLMs for NLG and NLP tasks means that generated output may not be reproducible in all cases.
2. LLM hallucinations, which are a common problem in NLG [105], were not addressed in this study - however we accept that this is a limitation.
3. Participant knowledge regarding threat modeling workshops of whom some may not have enough expertise to generate plausible attack scenarios. This was mitigated by using EoP card game and giving participants guidance how to formulate structured threat scenario descriptions.
4. Translation of the generated user scenarios from Estonian to English, which may have resulted in loss of information.
5. Model does not take into account the real-world context and likelihood of adversaries, but rather focuses on the theoretical capabilities of different adversary types.

5. Conclusion

The thesis presents a comprehensive approach to threat modeling for telepresence robots, considering various attack patterns, vulnerabilities, and components. This approach contributes to a deeper understanding of potential cyber(physical) attack paths, which is crucial for enhancing the security posture of TPR systems. Often times, visual representation of attack paths can be misleading and complex to understand as cognitive load increases with the complexity of the graph [67]. We hope however that a neutral, node based structure makes it easier to understand the flow of the attack paths.

Generated paths can be viewed individually, as seen on Figure 18, where the user can observe the metadata along with generated scenario. Numeric values such as likelihood and severity can be used to prioritize the most severe and likely paths for mitigation, while the generated scenarios can be used to give context and understanding of the attack path even to non-technical stakeholders. The integration of formal and informal techniques, including workshops, NLP and SSA, thesis showcases a multidimensional approach to attack path identification and analysis. This hybrid method bridges the gap between technical data and human interpretation, making the results accessible and actionable for cybersecurity professionals. By analyzing the results of the generated attack tree, we are able to answer RQ1, which is to identify attack paths that could be exploited through TPR system. In total, we identified 312 potential attack paths, which can be used to prioritize security measures and allocate resources more effectively to reduce the attack surface the most where the likelihood and severity is the highest. It is likely that majority of attacks will originate from the software domain, which constitutes 57% of the potential attack paths identified, indicating that efforts to secure the system should prioritize this area to effectively reduce the attack surface. Furthermore, attention should also be given to hardware (15%) and supply chain (14%) domains, as these also represent significant portions of the identified attack paths. Social engineering (8%) and physical security (6%) have lower proportions, but should not be neglected as they still contribute to the overall attack surface. These pattern domain results are beneficial for cybersecurity personnel to understand origins of possible attacks towards HCS through TPR system.

By analyzing target results of the attack tree, we are also able to answer RQ2, which is to identify the high value targets of the system. Based on the results there exists a possibility of 64% that an adversary might end up attacking the software of HCS, 28% for hardware and only 8% chance that an adversary might people (employees and patients). To best

protect HCS assets, the focus should be on securing the software of HCS as it's most likely to be attacked. Such results can be used to prioritize security measures and allocate resources more effectively to reduce the attack surface the most where the likelihood and severity is the highest.

Additional value aside from answering the research questions can be derived from the analysis of generated scenarios and token usage across different LLM models, provides a compelling insight into the efficiency and economic viability of using such models for cybersecurity threat modeling. The variation in token consumption and the associated costs among the models underline the importance of choosing the right model based on the specific requirements and budget constraints of a cybersecurity project. For instance, the gemini-1.0-pro model, while consuming a significant number of tokens, indicates a higher cost, which might not always correlate with better quality or relevance of the generated scenarios. Conversely, the gpt-3.5-turbo model showcased the most economical use of tokens, suggesting that effective scenario generation does not necessarily have to incur higher costs (see Figure 26).

Moreover, the semantic similarity analysis between user-generated and LLM-generated scenarios demonstrates a promising level of alignment, with scores indicating that the generated scenarios were not only realistic but closely mirrored human-generated scenarios (see Figure 27). This suggests that LLMs, when provided with well-structured prompts and adequate context, can produce relevant and realistic cybersecurity threat scenarios. Such capability is invaluable in enriching threat models with diverse and plausible attack scenarios, especially in the context of TPR where the potential attack surface might be vast and varied or even, unknown at the time.

Lastly, the focus on key components like microcontrollers, cameras, and motorized height control, and their frequent appearance in the generated attack paths, underscores the criticality of these components in telepresence robot security (see Figure 23). The analysis draws attention to potential vulnerabilities and emphasizes the necessity for targeted security measures to protect against exploits that could compromise the confidentiality, integrity, and availability of telepresence robots. In some cases, there might be configuration settings available for the user to change. For example, it might be in the administrator's best interest to disable the ability to use motorized height control if it's not needed. Or to disable the ability to use the robot's camera when the robot is not in use (e.g.: park the robot in a closed room) to prevent unauthorized surveillance. By looking at the most used components and their associated likelihood and severity, the administrator can make informed decisions on which functionalities to disable or enable to reduce the attack surface.

In conclusion, leveraging LLMs for NLP or NLG tasks presents a cost-effective, efficient, and versatile approach to enriching threat models. The integration of SSA further enhances the quality and relevance of the generated scenarios, whilst proposed modeling method provides a formal, structured approach to threat modeling of complex cyber(physical) systems. Threat modeling by analyzing attack patterns, system components, related weaknesses and known vulnerabilities of TPR systems can provide valuable insights into the potential attack paths and high-value targets. Enriching the threat model with diverse and plausible attack scenarios, where the potential attack surface might be vast and varied can provide a more meaningful understanding of the models output. This combined modeling technique not only aids in the anticipation of potential attack vectors but also facilitates a more comprehensive understanding of the cybersecurity landscape surrounding TPRs and where to direct resources for future security improvements.

References

[1] A. Rikalovic, N. Suzic, B. Bajic, and V. Piuri, "Industry 4.0 implementation challenges and opportunities: A technological perspective," *IEEE Systems Journal*, vol. 16, no. 2, pp. 2797–2810, 2021.

[2] M. Wehde, "Healthcare 4.0," *IEEE Engineering Management Review*, vol. 47, no. 3, pp. 24–28, 2019. DOI: 10.1109/EMR.2019.2930702.

[3] V. Kavidha, N. Gayathri, and S. R. Kumar, "Ai, iot and robotics in the medical and healthcare field," *AI and IoT-Based Intelligent Automation in Robotics*, pp. 165–187, 2021.

[4] J. T. Licardo, M. Domjan, and T. Orehovački, "Intelligent robotics—a systematic review of emerging technologies and trends," *Electronics*, vol. 13, no. 3, p. 542, 2024.

[5] Statista, *Worldwide medical robotics market size*, [Accessed: 20-04-2024]. URL: https://www.statista.com/statistics/1321270/worldwide-medical-robotics-market-size.

[6] A. Khan and Y. Anwar, "Robots in healthcare: A survey," in *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 2 1*, Springer, 2020, pp. 280–292.

[7] I.-B. Păvăloiu, A. Vasilățeanu, R. Popa, D. Scurtu, A. Hang, and N. Goga, "Healthcare robotic telepresence," in *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, IEEE, 2021, pp. 1–6.

[8] J. Leoste, K. Strömberg-Järvis, T. Robal, K. Marmor, K. Kangur, and A.-M. Rebane, "Testing scenarios for using telepresence robots in healthcare settings," *Computational and Structural Biotechnology Journal*, vol. 24, pp. 105–114, 2024.

[9] D. M. Hilty, K. Randhawa, M. M. Maheu, A. J. McKean, R. Pantera, M. C. Mishkind, and A. " Rizzo, "A review of telepresence, virtual reality, and augmented reality applied to clinical care," *Journal of Technology in Behavioral Science*, vol. 5, pp. 178–205, 2020.

[10] I. C. C. Center, *2023 internet crime report*, [Accessed: 14-04-2023]. URL: https://www.ic3.gov/Media/PDF/AnnualReport/2023_IC3Report.pdf, FBI, 2023.

[11] E. Fosch-Villaronga and T. Mahler, "Cybersecurity, safety and robots: Strengthening the link between cybersecurity and safety in the context of care robots," *Computer law & security review*, vol. 41, p. 105 528, 2021.

[12] V. Mayoral-Vilches, "Robot cybersecurity, a review," *International Journal of Cyber Forensics and Advanced Threat Investigations*, 2022.

[13] D. Giansanti and R. A. Gulino, "The cybersecurity and the care robots: A viewpoint on the open problems and the perspectives," in *Healthcare*, MDPI, vol. 9, 2021, p. 1653.

[14] A. Reindl, N. Rudigkeit, M. Ebers, M. Tröbinger, J. Elsner, and S. Haddadin, "Legal and technical considerations on unified, safe and data-protected haptic telepresence in healthcare," in *2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, IEEE, 2021, pp. 239–243.

[15] A. Rojas and S. Nørskov, "Interactions afforded by mobile telepresence robots in health care settings," in *HCI International 2023 Posters*, C. Stephanidis, M. Antona, S. Ntoa, and G. Salvendy, Eds., Cham: Springer Nature Switzerland, 2023, pp. 138–145, ISBN: 978-3-031-35992-7.

[16] K. Youssef, S. Said, S. Al Kork, and T. Beyrouthy, "Telepresence in the recent literature with a focus on robotic platforms, applications and challenges," *Robotics*, vol. 12, no. 4, p. 111, 2023.

[17] J.-P. A. Yaacoub, H. N. Noura, O. Salman, and A. Chehab, "Robotics cyber security: Vulnerabilities, attacks, countermeasures, and recommendations," *International Journal of Information Security*, pp. 1–44, 2022.

[18] C. Cerrudo and L. Apa, "Hacking robots before skynet," 2017.

[19] G. Lacava, A. Marotta, F. Martinelli, A. Saracino, A. La Marra, E. Gil-Uriarte, and V. M. Vilches, "Cybsersecurity issues in robotics.," *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, vol. 12, no. 3, pp. 1–28, 2021.

[20] A. Botta, S. Rotbei, S. Zinno, and G. Ventre, "Cyber security of robots: A comprehensive survey," *Intelligent Systems with Applications*, vol. 18, p. 200 237, 2023, ISSN: 2667-3053. DOI: https://doi.org/10.1016/j.iswa.2023.200237.

[21] E. Fosch Villaronga and T. Mahler, "Cybersecurity, safety and robots: Strengthening the link between cybersecurity and safety in the context of care robots," *Computer Law & Security Review*, vol. 41, p. 105 528, Jul. 2021. DOI: 10.1016/j.clsr.2021.105528.

[22] K. M. A. Yousef, A. AlMajali, S. Abu Ghalyon, W. Dweik, and B. J. Mohd, "Analyzing cyber-physical threats on robotic platforms," *SENSORS*, vol. 18, no. 5, 2018. DOI: 10.3390/s18051643.

[23] C. S. Kruse, B. Frederick, T. Jacobson, and D. K. Monticone, "Cybersecurity in healthcare: A systematic review of modern threats and trends," *Technology and Health Care*, vol. 25, no. 1, pp. 1–10, 2017.

[24] A. Gupta, A. Singh, D. Bharadwaj, and A. K. Mondal, "Humans and robots: A mutually inclusive relationship in a contagious world," *International Journal of Automation and Computing*, vol. 18, pp. 185–203, 2021.

[25] J. Leoste, A. Talisainen, J. Pöial, K. Kangur, T. Kasuk, and J. Parts, "Exploring telepresence robots in higher education: A case study," in *Advances in Information and Communication*, K. Arai, Ed., Cham: Springer Nature Switzerland, 2024, pp. 694–704, ISBN: 978-3-031-53960-2.

[26] *Query 1 ieee xplore: Telepresence and (robot or robots) and (medical or hospital or healthcare)*, [Accessed: 16-03-2024]. [Online]. Available: https://ieeexplore.ieee.org/search/searchresult.jsp?queryText=telepresence%20AND%20(robot%20OR%20robots)%20AND%20(medical%20OR%20hospital%20OR%20healthcare)&highlight=true&returnType=SEARCH&matchPubs=true&ranges=2014_2024_Year&returnFacets=ALL&refinements=ContentType:Conferences&refinements=ContentType:Journals.

[27] *Query 1 scopus: Telepresence and (robot or robots) and (medical or hospital or healthcare)*, [Accessed: 16-03-2024]. [Online]. Available: https://www.scopus.com/results/results.uri?sort=plf-f&src=s&st1=cybersecurity+AND+%28robots+OR+telepresence%29&sot=b&sdt=cl&sl=57&s=TITLE-ABS-KEY%28telepresence+AND+%28robot+OR+robots%29+AND+%28medical+OR+hospital+OR+healthcare%29%29&yearFrom=2014&yearTo=2024.

[28] *Query 1 springer: Telepresence and (robot or robots) and (medical or hospital or healthcare)*, [Accessed: 16-03-2024]. [Online]. Available: https://link.springer.com/search?query=telepresence+AND+%28robot+AND+OR+AND+robots%29+AND+AND+AND+%28medical+AND+OR+AND+hospital+AND+OR+AND+healthcare%29&facet-end-year=2024&showAll=true&date-facet-mode=between&facet-start-year=2014.

[29] *Query 1 web of science: Telepresence and (robot or robots) and (medical or hospital or healthcare)*, [Accessed: 16-03-2024]. [Online]. Available: https://www.webofscience.com/wos/woscc/summary/129cba6d-6b78-4989-96b9-b01596bfe484-d304f603/relevance/1.

[30] *Query 2 ieee xplore: Telepresence and (robot or robots) and (cybersecurity or cyber)*, [Accessed: 16-03-2024]. [Online]. Available: https://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=telepresence%20AND%20(%20robot%20OR%20robots%20)%20AND%20(%20cybersecurity%20OR%20cyber%20).

[31] *Query 2 scopus: Telepresence and (robot or robots) and (cybersecurity or cyber)*, [Accessed: 16-03-2024]. [Online]. Available: https://www.scopus.com/results/results.uri?sort=plf-f&src=s&st1=cybersecurity+AND+%28robots+OR+telepresence%29&sot=b&sdt=b&sl=57&s=TITLE-ABS-KEY%28telepresence+AND+%28+robot+OR+robots+%29+AND+%28+cybersecurity+OR+cyber+%29%29&yearFrom=2014&yearTo=2024.

[32] *Query 2 springer: Telepresence and (robot or robots) and (cybersecurity or cyber)*, [Accessed: 16-03-2024]. [Online]. Available: https://link.springer.com/search?query=%E2%80%99telepresence+AND+%28robot+OR+robots%29+AND+%28cybersecurity+OR+cyber%29&facet-end-year=2024&showAll=true&date-facet-mode=between&facet-start-year=2014.

[33] *Query 2 web of science: Telepresence and (robot or robots) and (cybersecurity or cyber)*, [Accessed: 16-03-2024]. [Online]. Available: https://www.webofscience.com/wos/woscc/summary/f12b3d33-37c2-425b-9649-d76f8b10a9fe-d3057f85/relevance/1.

[34] V. D. B. I. Report, *2023 data breach investigations report*, [Accessed: 27-01-2024]. URL: https://www.verizon.com/business/resources/reports/2023-data-breach-investigations-report-dbir.pdf, 2023.

[35] European Union Agency for Cybersecurity (ENISA), *Enisa threat landscape: Health sector*, M. Theocharidou and I. Lella, Eds., https://www.enisa.europa.eu, 2023. DOI: 10.2824/163953.

[36] W. E. Forum, *The global risks report 2023*, [Accessed: 14-03-2023]. URL: https://www3.weforum.org/docs/WEF_Global_Risks_Report_2023.pdf, World Economic Forum, 2023.

[37] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, "Stride-based threat modeling for cyber-physical systems," in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, IEEE, 2017, pp. 1–6.

[38] W. Xiong and R. Lagerström, "Threat modeling–a systematic literature review," *Computers & security*, vol. 84, pp. 53–69, 2019.

[39] P. Johnson, R. Lagerström, and M. Ekstedt, "A meta language for threat modeling and attack simulations," in *Proceedings of the 13th International Conference on Availability, Reliability and Security*, 2018, pp. 1–8.

[40] S. Barnum, "Standardizing cyber threat intelligence information with the structured threat information expression (stix)," *Mitre Corporation*, vol. 11, pp. 1–22, 2012.

[41] I. Stellios, P. Kotzanikolaou, and C. Grigoriadis, "Assessing iot enabled cyber-physical attack paths against critical systems," *Computers & Security*, vol. 107, p. 102 316, 2021, ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2021.102316.

[42] J. Holland, L. Kingston, C. McCarthy, E. Armstrong, P. O'Dwyer, F. Merz, and M. McConnell, "Service robots in the healthcare sector," *Robotics*, vol. 10, no. 1, p. 47, 2021.

[43] A. A. Morgan, J. Abdi, M. A. Syed, G. E. Kohen, P. Barlow, and M. P. Vizcaychipi, "Robots in healthcare: A scoping review," *Current Robotics Reports*, vol. 3, no. 4, pp. 271–280, 2022.

[44] K. A. R. Carranza, N. J. B. Day, L. M. S. Lin, A. R. Ponce, W. R. O. Reyes, A. C. Abad, and R. G. Baldovino, "Akibot: A telepresence robot for medical teleconsultation," in *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)*, 2018, pp. 1–4. DOI: 10.1109/HNICEM.2018.8666283.

[45] M. Wang, C. Pan, and P. K. Ray, "Technology entrepreneurship in developing countries: Role of telepresence robots in healthcare," *IEEE Engineering Management Review*, vol. 49, no. 1, pp. 20–26, 2021.

[46] I. Kim, A. Nepomuceno, S. Jamison, J. Michel, and T. Kesavadas, "Extensive simulation of human-robot interaction for critical care telemedicine," in *2022 Annual Modeling and Simulation Conference (ANNSIM)*, IEEE, 2022, pp. 329–340.

[47] F. Naseer, M. N. Khan, and A. Altalbe, "Telepresence robot with drl assisted delay compensation in iot-enabled sustainable healthcare environment," *Sustainability*, vol. 15, no. 4, p. 3585, 2023.

[48] F. J. R. Lera, C. F. Llamas, Á. M. Guerrero, and V. M. Olivera, "Cybersecurity of robotics and autonomous systems: Privacy and safety," *Robotics-legal, ethical and socioeconomic impacts*, 2017.

[49] G. Lacava, A. Marotta, F. Martinelli, A. Saracino, A. La Marra, E. Gil-Uriarte, and V. M. Vilches, "Cybsersecurity issues in robotics.," *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, vol. 12, no. 3, pp. 1–28, 2021.

[50] J. Rajamäki and M. Järvinen, "Exploring care robots' cybersecurity threats from care robotics specialists' point of view," in *European Conference on Cyber Warfare and Security*, vol. 21, 2022, pp. 231–238.

[51] S. Morante, J. G. Victores, and C. Balaguer, "Cryptobotics: Why robots need cyber safety," *Frontiers in Robotics and AI*, vol. 2, p. 23, 2015.

[52] J. Rajamäk and M. Järvinen, "Exploring care robots' cybersecurity threats from care robotics specialists' point of view," in *Proceedings of the 21th European Conference on Cyber Warfare and Security (ECCWS 2022)*, Academic Conferences International Limited, 2022.

[53] S. O. Oruma, Y. Z. Ayele, F. Sechi, and H. Rødsethol, "Security aspects of social robots in public spaces: A systematic mapping study," *Sensors*, vol. 23, no. 19, p. 8056, 2023.

[54] S. O. Oruma, M. Sánchez-Gordón, R. Colomo-Palacios, V. Gkioulos, and J. K. Hansen, "A systematic review on social robots in public spaces: Threat landscape and attack surface," *Computers*, vol. 11, no. 12, p. 181, 2022.

[55] S. Tweedian, *This hilarious video shows a robot escaping from a locked room*, [Accessed: 18-03-2024] URL: https://www.businessinsider.in/this-hilarious-video-shows-a-robot-escaping-from-a-locked-room/articleshow/45263766.cms.

[56] C. Cimpanu, *Flaws in telepresence robots allow hackers access to pictures, video feeds*, [Accessed: 08-03-2024]. URL: https://www.zdnet.com/article/flaws-in-telepresence-robots-allow-hackers-access-to-pictures-video-feeds.

[57] Cybersecurity and I. S. Agency, *Vecna vgo robot (update a)*, [Accessed: 08-03-2024]. URL: https://www.cisa.gov/news-events/ics-advisories/icsa-18-114-01.

[58] M. Mimoso, *Telepresence robots patched against data leaks*, [Accessed: 08-03-2024]. URL: https://threatpost.com/telepresence-robots-patched-against-data-leaks/124257/.

[59] S. M. Kerner, *Rapid7 reveals telepresence robot iot vulnerabilities*, [Accessed: 09-03-2024]. URL: https://www.eweek.com/security/rapid7-reveals-telepresence-robot-iot-vulnerabilities/.

[60] T. Sandle, *Risk vulnerabilities with telepresence robots*, [Accessed: 09-03-2024]. URL: https://www.digitaljournal.com/tech-science/risk-vulnerabilities-with-telepresence-robots/article/535143.

[61] I. IOActive, *Ioactive conducts first-ever ransomware attack on robots at kaspersky security analyst summit 2018*, Press release, [Accessed: 24-03-2024]. URL: https://ioactive.com/article/ioactive-conducts-first-ever-ransomware-attack-on-robots-at-kaspersky-security-analyst-summit-2018/.

[62] M. Burgess, *Ethical hackers have turned this robot into a stabbing machine*, Wired, [Accessed: 24-03-2024]. URL: https://www.wired.com/story/hacked-robots-pepper-nao-alpha-2-stab-screwdriver/.

[63] C. Hankin, P. Malacaria, *et al.*, "Attack dynamics: An automatic attack graph generation framework based on system topology, capec, cwe, and cve databases," *Computers & Security*, vol. 123, p. 102 938, 2022.

[64] A. Brazhuk, "Towards automation of threat modeling based on a semantic model of attack patterns and weaknesses," *arXiv preprint arXiv:2112.04231*, 2021.

[65] N. I. of Standards and Technology, *Attack tree - glossary of key information security terms*, [Accessed: 18-03-2024]. URL: https://csrc.nist.gov/glossary/term/attack_tree.

[66] A. P. Moore, R. J. Ellison, R. C. Linger, *et al.*, "Attack modeling for information security and survivability," 2001.

[67] H. S. Lallie, K. Debattista, and J. Bal, "A review of attack graph and attack tree visual syntax in cyber security," *Computer Science Review*, vol. 35, p. 100 219, 2020.

[68] Kaspersky, *Attack vector*, [Accessed: 24-03-2024]. URL: https://encyclopedia.kaspersky.com/glossary/attack-vector/.

[69] J. Selin, "Evaluation of threat modeling methodologies," 2019.

[70] W. Xiong, E. Legrand, O. Åberg, and R. Lagerström, "Cyber security threat modeling based on the mitre enterprise att&ck matrix," *Software and Systems Modeling*, vol. 21, no. 1, pp. 157–177, 2022.

[71] N. I. of Standards and Technology, *Attack surface - glossary of key information security terms*, [Accessed: 18-03-2024]. URL: https://csrc.nist.gov/glossary/term/attack_surface.

[72] P. K. Manadhata and J. M. Wing, "An attack surface metric," *IEEE Transactions on Software Engineering*, vol. 37, no. 3, pp. 371–386, 2010.

[73] M. Maidl, G. Münz, S. Seltzsam, M. Wagner, R. Wirtz, and M. Heisel, "Model-based threat modeling for cyber-physical systems: A computer-aided approach," in *Software Technologies: 15th International Conference, ICSOFT 2020, Online Event, July 7–9, 2020, Revised Selected Papers 15*, Springer, 2021, pp. 158–183.

[74] W. Duo, M. Zhou, and A. Abusorrah, "A survey of cyber attacks on cyber physical systems: Recent advances and challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 5, pp. 784–800, 2022.

[75] *Attack path analysis*, [Accessed: 22-03-2024]. URL: https://www.rapid7.com/fundamentals/attack-path-analysis.

[76] W. Peng, S. Yao, and J. Chen, "Recognizing intrusive intention and assessing threat based on attack path analysis," in *2009 International Conference on Multimedia Information Networking and Security*, IEEE, vol. 2, 2009, pp. 450–453.

[77] H. Xie, K. Lv, and C. Hu, "An improved monte carlo graph search algorithm for optimal attack path analysis," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018, pp. 307–315. DOI: 10.1109/TrustCom/BigDataSE.2018.00054.

[78] H. Zhang, S. Kang, and Y. Li, "Visual construction algorithm of attack path based on medical sensor networks," in *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*, 2018, pp. 775–779. DOI: 10.1109/IICSPI.2018.8690460.

[79] Y. Chen, B. Boehm, and L. Sheppard, "Value driven security threat modeling based on attack path analysis," in *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, IEEE, 2007, 280a–280a.

[80] S. Hussain, H. Erwin, and P. Dunne, "Threat modeling using formal methods: A new approach to develop secure web applications," in *2011 7th International Conference on Emerging Technologies*, IEEE, 2011, pp. 1–5.

[81] D. Xu, M. Tu, M. Sanford, L. Thomas, D. Woodraska, and W. Xu, "Automated security test generation with formal threat models," *IEEE transactions on dependable and secure computing*, vol. 9, no. 4, pp. 526–540, 2012.

[82] E. G. Spanakis, S. Bonomi, S. Sfakianakis, G. Santucci, S. Lenti, M. Sorella, F. D. Tanasache, A. Palleschi, C. Ciccotelli, V. Sakkalis, and S. Magalini, "Cyber-attacks and threats for healthcare – a multi-layer thread analysis," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2020, pp. 5705–5708. DOI: 10.1109/EMBC44109.2020.9176698.

[83] D. Xu and K. E. Nygard, "Threat-driven modeling and verification of secure software using aspect-oriented petri nets," *IEEE transactions on software engineering*, vol. 32, no. 4, pp. 265–278, 2006.

[84] MITRE, *Common vulnerabilities and exposures*, [Accessed: 22-03-2024]. URL: https://cve.mitre.org, 2024.

[85] MITRE, *Common weakness enumeration*, [Accessed: 22-03-2024]. URL: https://cwe.mitre.org, 2024.

[86] MITRE, *Common attack pattern enumeration and classification*, [Accessed: 22-03-2024]. URL: https://capec.mitre.org, 2024.

[87] N. Messe, V. Chiprianov, N. Belloir, J. El-Hachem, R. Fleurquin, and S. Sadou, "Asset-oriented threat modeling," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, IEEE, 2020, pp. 491–501.

[88] A. Honkaranta, T. Leppänen, and A. Costin, "Towards practical cybersecurity mapping of stride and cwe—a multi-perspective approach," in *2021 29th Conference of Open Innovations Association (FRUCT)*, IEEE, 2021, pp. 150–159.

[89] J. Simonjan, S. Taurer, and B. Dieber, "A generalized threat model for visual sensor networks," *Sensors*, vol. 20, no. 13, p. 3629, 2020.

[90] S. Krishnan, "A hybrid approach to threat modelling," 2017. DOI: 10.13140/RG.2.2.33303.88486.

[91] S. Christey, J. Kenderdine, J. Mazella, and B. Miles, "Common weakness enumeration," *Mitre Corporation*, 2013.

[92] C. Watson and T. Zaw, *OWASP automated threat handbook: Web applications*. OWASP Foundation, 2018.

[93] Y. Wu, I. Bojanova, and Y. Yesha, "They know your weaknesses–do you?: Reintroducing common weakness enumeration," *CrossTalk*, vol. 45, 2015.

[94] J. Glyder, A. K. Threatt, R. Franks, L. Adams, and G. Stoker, "Some analysis of common vulnerabilities and exposures (cve) data from the national vulnerability database (nvd)," in *Proceedings of the Conference on Information Systems Applied Research ISSN*, vol. 2167, 2021, p. 1508.

[95] K. Kanakogi, H. Washizaki, Y. Fukazawa, S. Ogata, T. Okubo, T. Kato, H. Kanuka, A. Hazeyama, and N. Yoshioka, "Tracing cve vulnerability information to capec attack patterns using natural language processing techniques," *Information*, vol. 12, no. 8, p. 298, 2021.

[96] S. Zhang, D. Caragea, and X. Ou, "An empirical study on using the national vulnerability database to predict software vulnerabilities," in *Database and Expert Systems Applications: 22nd International Conference, DEXA 2011, Toulouse, France, August 29-September 2, 2011. Proceedings, Part I 22*, Springer, 2011, pp. 217–231.

[97] G. Kim, P. Baldi, and S. McAleer, "Language models can solve computer tasks," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[98] M. Gupta, C. Akiri, K. Aryal, E. Parker, and L. Praharaj, "From chatgpt to threatgpt: Impact of generative ai in cybersecurity and privacy," *IEEE Access*, 2023.

[99] H. Hibshi, S. T. Jones, and T. D. Breaux, "A systemic approach for natural language scenario elicitation of security requirements," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 6, pp. 3579–3591, 2021.

[100] F. Perrina, F. Marchiori, M. Conti, and N. V. Verde, "Agir: Automating cyber threat intelligence reporting with natural language generation," in *2023 IEEE International Conference on Big Data (BigData)*, IEEE, 2023, pp. 3053–3062.

[101] Y. Yigit, W. J. Buchanan, M. G. Tehrani, and L. Maglaras, "Review of generative ai methods in cybersecurity," *arXiv preprint arXiv:2403.08701*, 2024.

[102] J.-Y. Yao, K.-P. Ning, Z.-H. Liu, M.-N. Ning, and L. Yuan, "Llm lies: Hallucinations are not bugs, but features as adversarial examples," *arXiv preprint arXiv:2310.01469*, 2023.

[103] L. Weidinger, J. Mellor, M. Rauh, C. Griffin, J. Uesato, P.-S. Huang, M. Cheng, M. Glaese, B. Balle, A. Kasirzadeh, *et al.*, "Ethical and social risks of harm from language models," *arXiv preprint arXiv:2112.04359*, 2021.

[104] E. Crothers, N. Japkowicz, and H. L. Viktor, "Machine-generated text: A comprehensive survey of threat models and detection methods," *IEEE Access*, 2023.

[105] Z. Xu, S. Jain, and M. Kankanhalli, "Hallucination is inevitable: An innate limitation of large language models," *arXiv preprint arXiv:2401.11817*, 2024.

[106] I. Elsharef, Z. Zeng, and Z. Gu, "Facilitating threat modeling by leveraging large language models,"

[107] K. Jones, E. Altuncu, V. N. Franqueira, Y. Wang, and S. Li, "A comprehensive survey of natural language generation advances from the perspective of digital deception," *arXiv preprint arXiv:2208.05757*, 2022.

[108] P. Mehta and S. Pandya, "A review on sentiment analysis methodologies, practices and applications," *International Journal of Scientific and Technology Research*, vol. 9, no. 2, pp. 601–609, 2020.

[109] M. U. Hadi, R. Qureshi, A. Shah, M. Irfan, A. Zafar, M. B. Shaikh, N. Akhtar, J. Wu, S. Mirjalili, *et al.*, "Large language models: A comprehensive survey of its applications, challenges, limitations, and future prospects," *Authorea Preprints*, 2023.

[110] L. Ignaczak, G. Goldschmidt, C. A. D. Costa, and R. D. R. Righi, "Text mining in cybersecurity: A systematic literature review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 7, pp. 1–36, 2021.

[111] K. Hayes, "Comparative evaluation of sentiment analysis methods," 2022.

[112] A. Gołębiowska, W. Jakubczak, D. Prokopowicz, and R. Jakubczak, "Cybersecurity of business intelligence analytics based on the processing of large sets of information with the use of sentiment analysis and big data," *European Research Studies Journal*, vol. 24, no. 4, 2021.

[113] B. Thapa, "Sentiment analysis of cybersecurity content on twitter and reddit," *arXiv preprint arXiv:2204.12267*, 2022.

[114] D. Lundquist, K. Zhang, and A. Ouksel, "Ontology-driven cyber-security threat assessment based on sentiment analysis of network activity data," in *2014 International Conference on Cloud and Autonomic Computing*, IEEE, 2014, pp. 5–14.

[115] S. Silvestri, S. Islam, D. Amelin, G. Weiler, S. Papastergiou, and M. Ciampi, "Cyber threat assessment and management for securing healthcare ecosystems using natural language processing," *International Journal of Information Security*, pp. 1–20, 2023.

[116] O. D. Okey, E. U. Udo, R. L. Rosa, D. Z. Rodríguez, and J. H. Kleinschmidt, "Investigating chatgpt and cybersecurity: A perspective on topic modeling and sentiment analysis," *Computers & Security*, vol. 135, p. 103 476, 2023.

[117] M. Alawida, S. Mejri, A. Mehmood, B. Chikhaoui, and O. Isaac Abiodun, "A comprehensive study of chatgpt: Advancements, limitations, and ethical considerations in natural language processing and cybersecurity," *Information*, vol. 14, no. 8, p. 462, 2023.

[118] X. Tong, B. Jin, Z. Lin, B. Wang, and T. Yu, "Cpsdbench: A large language model evaluation benchmark and baseline for chinese public security domain," *arXiv preprint arXiv:2402.07234*, 2024.

[119] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[120] A. Buscemi and D. Proverbio, "Chatgpt vs gemini vs llama on multilingual sentiment analysis," *arXiv preprint arXiv:2402.01715*, 2024.

[121] N. Rane, S. Choudhary, and J. Rane, "Gemini versus chatgpt: Applications, performance, architecture, capabilities, and implementation," *Performance, Architecture, Capabilities, and Implementation (February 13, 2024)*, 2024.

[122] M. I. Hossen, A. Islam, F. Anowar, E. Ahmed, and M. M. Rahman, "Generating cyber threat intelligence to discover potential security threats using classification and topic modeling," in *Cyber Security Using Modern Technologies*, CRC Press, pp. 141–153.

[123] D. M. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil, "Universal sentence encoder," *arXiv preprint arXiv:1803.11175*, 2018. DOI: 10.48550/arXiv.1803.11175.

[124] M. Dehghani, R. Boghrati, K. Man, J. Hoover, S. I. Gimbel, A. Vaswani, J. D. Zevin, M. H. Immordino-Yang, A. S. Gordon, A. Damasio, *et al.*, "Decoding the neural representation of story meanings across languages," *Human brain mapping*, vol. 38, no. 12, pp. 6096–6106, 2017.

[125] J. Rabelo, M.-Y. Kim, R. Goebel, M. Yoshioka, Y. Kano, and K. Satoh, "A summary of the coliee 2019 competition," in *New Frontiers in Artificial Intelligence: JSAI-isAI International Workshops, JURISIN, AI-Biz, LENLS, Kansei-AI, Yokohama, Japan, November 10–12, 2019, Revised Selected Papers 10*, Springer, 2020, pp. 34–49.

[126] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. St. John, N. Constant, M. Guajardo-Céspedes, S. Yuan, C. Tar, *et al.*, *Universal sentence encoder*, [Accessed: 07-03-2024]. URL: https://www.kaggle.com/models/google/universal-sentence-encoder/frameworks/tensorFlow1/variations/lite/versions/2.

[127] N. Sun, M. Ding, J. Jiang, W. Xu, X. Mo, Y. Tai, and J. Zhang, "Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives," *IEEE Communications Surveys & Tutorials*, 2023.

[128] J. Parts, *Modeling project*, [Accessed: 16-03-2024]. URL: https://gitlab.cs.ttu.ee/jopart/model.

[129] A. Shostack, "Elevation of privilege: Drawing developers into threat modeling," in *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 14)*, 2014.

[130] E. W. Burger, M. D. Goodman, P. Kampanakis, and K. A. Zhu, "Taxonomy model for cyber threat intelligence information exchange technologies," in *Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security*, 2014, pp. 51–60.

[131] S. Islam, S. Papastergiou, E.-M. Kalogeraki, and K. Kioskli, "Cyberattack path generation and prioritisation for securing healthcare systems," *Applied Sciences*, vol. 12, no. 9, p. 4443, 2022.

[132] J. Parts, *Semantic similarity analysis playbook*, [Accessed: 08-03-2024]. URL: https://gitlab.cs.ttu.ee/jopart/thesis_ssa/-/blob/master/src/playbook.ipynb.

[133] D. Soyusiawaty and Y. Zakaria, "Book data content similarity detector with cosine similarity (case study on digilib. uad. ac. id)," in *2018 12th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, IEEE, 2018, pp. 1–6.

[134] J. Parts, *Semantic similarity analysis for thesis data between user and generated threat scenarios*, [Accessed: 08-03-2024]. URL: https://gitlab.cs.ttu.ee/jopart/thesis_ssa/-/blob/master/src/semantic_similarity_analysis_results.csv.

[135] J. Parts, *User scenarios*, [Accessed: 16-03-2024]. URL: https://gitlab.cs.ttu.ee/jopart/model/-/blob/master/src/main/resources/results/results_user_scenarios.csv.

[136] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "Mitre att&ck: Design and philosophy," in *Technical report*, The MITRE Corporation, 2018.

[137] A. F. Brantly, "Cyber actions by state actors: Motivation and utility," *International Journal of Intelligence and CounterIntelligence*, vol. 27, no. 3, pp. 465–484, 2014.

[138] M. B. Barcena and C. Wueest, "Insecurity in the internet of things," *Security response, symantec*, vol. 20, 2015.

[139] N. Nissim, R. Yahalom, and Y. Elovici, "Usb-based attacks," *Computers & Security*, vol. 70, pp. 675–688, 2017.

[140] J. Parts, *Generated scenarios*, [Accessed: 16-03-2024]. URL: https://gitlab.cs.ttu.ee/jopart/model/-/blob/master/src/main/resources/results/results_scenarios.csv.

[141] *Google ai gemini pricing*, [Accessed: 16-03-2024]. URL: https://ai.google.dev/pricing.

[142] *Ibm watson machine learning pricing*, [Accessed: 16-03-2024]. URL: https://dataplatform.cloud.ibm.com/docs/content/wsj/getting-started/wml-plans.html?context=cpdaas.

[143] *Openai pricing*, [Accessed: 16-03-2024]. URL: https://openai.com/pricing.

Appendix 1 – Non-Exclusive License for Reproduction and Publication of a Graduation Thesis[1]

I Joosep Parts

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Attack Pattern Assessment of Telepresence Robots in Healthcare Systems Context", supervised by Kaido Kikkas and Janika Leoste
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

12.05.2024

[1]The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 – Threat modeling workshop questionnaire (English)

Dear participant!

We would like to ask **your help in generating threat statements** - what could go wrong with the use of telepresence robot systems? Your answers will help us validate computer generated attack paths and help us understand current cybersecurity situation regarding telepresence robots. **Your answers are anonymous, answering the questions will take 3-4 minutes. Please fill in your information and read the example before answering.**

Contacts:

Experimenter	**Supervisor**	**Supervisor**
Joosep Parts, Cybersecurity MSc student jopart@taltech.ee	Kaido Kikkas, Ph.D kaido.kikkas@taltech.ee	Janika Leoste, Ph.D janika.leoste@taltech.ee

Your information:

[1] Consent: by checking the following box I give my consent to the above person(s) for processing my answers in the study: ☐

[2] Gender: I am female: ☐ male: ☐ other: ☐ **[3]** My age is: ☐ **[4]** Card number: ☐

We would like you to step into the shoes of potential adversary. You've been given a STRIDE playing card which will help you generate some ideas. Think about cybersecurity incidents you've heard about and try to generate an idea how to harm telepresence robot system. Then use structured threat statement(look below) to write down your answer.

Example:

"An attacker could exploit **[weakness]** in **[component]** to **[action]**, leading to **[consequence]**" *"An attacker could exploit insufficient authentication in the robot's remote access protocol to gain unauthorized access, potentially leading to data compromise.*

Your answer in free form *(please keep it readable)***:**

Appendix 3 – Threat modeling workshop questionnaire (Estonian)

Hea osaleja!

Soovime paluda **teie abi küberohtude genereerimisel** - mis võib valesti minna kaugosalusroboti süsteemide kasutamisel? Teie vastused aitavad meil valideerida arvutiga genereeritud rünnakute suundasid ja mõista paremini praegust kaugosalusrobotite küberjulgeoleku seisu. **Teie vastused on anonüümsed, küsimustele vastamine võtab aega 3-4 minutit. Palun täitke oma andmed ja lugege näidet enne vastamist.**

Kontaktid:

Läbiviija	**Juhendaja**	**Juhendaja**
Joosep Parts, Küberkaitse MSc tudeng jopart@taltech.ee	Kaido Kikkas, Ph.D kaido.kikkas@taltech.ee	Janika Leoste, Ph.D janika.leoste@taltech.ee

Teie info:

[1] Nõusolek: järgmist kasti märkides annan nõusoleku ülaltoodud isiku(te)le minu vastuste töötlemiseks uuringus: ☐

[2] Sugu: olen naine: ☐ mees: ☐ muu: ☐ **[3]** Minu vanus: ☐ **[4]** Kaardi number: ☐

Soovime, et te astuksite potentsiaalse vaenlase kingadesse. Teile on antud STRIDE mängukaart, mis aitab teil mõned mõtted luua. Mõelge küberohtudest, millest olete kuulnud ja proovige kirjeldada enda stsenaariumi mis võiks kahjustada kaugosalusroboti süsteemi. Seejärel kasutage struktureeritud ohu kirjeldust(vt. allpool) oma vastuse kirjutamiseks.

Näide:

"Ründaja võib ära kasutada **[nõrkust] [komponendis] [tegevuse]** sooritamiseks, mis võib viia **[tagajärjeni]**"

"Ründaja võib ära kasutada ebapiisavat isikutuvastust roboti kaugjuurdepääsu protokollis, et saada volitamata juurdepääs, mis võib viia andmete kaoni."

Teie vastus vabas vormis *(palun loetavalt kirjutada)*:

Appendix 4 – Algorithm for generating attack paths

Algorithm 1: Generating attack paths.

Data: Threat Model $\mathcal{M}$ loaded with sets of Adversaries $\mathcal{A}$, Attack Patterns $\mathcal{P}$, Domains $\mathcal{D}$, Components $\mathcal{C}$, and Targets $\mathcal{T}$

Result: Augmented set of attack paths in $\mathcal{M}$

Initialize an empty list $\mathcal{AP} = \emptyset$;
forall $a \in \mathcal{A}$ **do**
  forall $p \in \mathcal{P}$ **do**
    forall $d \in \mathcal{D}$ **do**
      forall $c \in \mathcal{C}$ **do**
        forall $t \in \mathcal{T}$ **do**
          Let $path$ be a new Path object;
          Set $path$ as valid;
          Assign $a \mapsto path$.adversary;
          Assign $p \mapsto path$.attackPattern;
          Assign $d \mapsto path$.domain;
          Assign $c \mapsto path$.component;
          Assign $t \mapsto path$.target;
          Compute $f_{\text{severity}}(path, p)$;
          Compute $f_{\text{likelihood}}(path, p)$;
          Append $path \rightarrow \mathcal{AP}$;
Set $\mathcal{M}$.attackPaths $\leftarrow \mathcal{M}$.attackPaths $\cup \mathcal{AP}$;

Source code can be found at gitlab.cs.ttu.ee [128].

Appendix 5 – Algorithm for evaluating attack paths

Algorithm 2: Attack Path Evaluation

Input: Threat Model Data, CVE Catalog, Attack Pattern Catalog, Weakness Catalog
Output: Validated Set of Attack Paths

```
1  Function Generate:
       Data: Initial sets from Catalogs
       Result: {𝒜𝒫} where every path in set is valid path
2      foreach path in {𝒜𝒫} do
3          if evaluateDomain(path) is false then
4              path.setValid(false)
5              path.setInvalidReason("Pattern not in the domain")
6          if evaluateTarget(path) is false then
7              path.setValid(false)
8              path.setInvalidReason("Target is not in the domain")
9          if evaluateRelatedCWE(path) is false then
10             path.setValid(false)
11             path.setInvalidReason("Pattern does not have weakness")
12         if evaluateRelatedCVE(path) is false then
13             path.setValid(false)
14             path.setInvalidReason("CWE does not map to any CVE")
15         if evaluateAdversary(path) is false and path.isValid() then
16             path.setValid(false)
17         else
18             path.setValid(true)
19     Function Normalize:
20         Normalize and round values for each attribute in paths
21     Save valid paths to the database
```

Source code can be found at gitlab.cs.ttu.ee [128].

Appendix 6 – Algorithm for semantic similarity analysis

Algorithm 3: Semantic Similarity Analysis

Data: Scenario Dataframe S, User Scenario Dataframe U, Universal Sentence Encoder E
Result: Similarity Analysis Result File
Input : S, U, E
Output : Semantic Similarity Results CSV File

```
Load S from 'scenario_export.csv';
Load U from 'user_scenario_export.csv';
for each response r in S do
    Embed r using E to get embeddings e_r;
for each scenario s in U do
    Embed s using E to get embeddings e_s;
Initialize an empty matrix M to store similarities;
for each embedding e_r in S do
    for each embedding e_s in U do
        Compute cosine similarity c between e_r and e_s;
        Adjust c to be in the range [0,1];
        Store c in M;
Create a results dataframe R from M;
Sort R in descending order based on similarity;
Save R into semantic_similarity_analysis_results.csv;
```

Source code can be found at gitlab.cs.ttu.ee [134].