Security Requirements Engineering: A Framework for Cyber-Physical Systems

DISSERTATION

zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

durch die Fakultät für Wirtschaftswissenschaften der
Universität Duisburg-Essen, Campus Essen

vorgelegt von

Shafiq ur Rehman
geboren in Karachi, Pakistan

Betreuer: Prof. Dr. Volker Gruhn
Lehrstuhl für Software Engineering, insb. mobile Anwendungen
Institut für Informatik und Wirtschaftsinformatik

Essen, 2020

Gutachter:

1. Gutachter: Prof. Dr. Volker Gruhn

2. Gutachter: Prof. Dr. Matthias Book

Tag der mündlichen Prüfung: 14.01.2020

DOI: 10.17185/duepublico/71232
URN: urn:nbn:de:hbz:464-20200129-102905-3

Abstract

In present day software development industry, cyber-physical systems are gaining much attention from researchers and practitioners due to their high impact on the world's economy. These systems are considered as hallmarks of the modern age of computing power integrated with physical systems. With the rising use and importance of cyber-physical systems, organizations have come to terms with the importance of security in these systems. Therefore, security requirements are a significant part of cyber-physical systems, but there is a lack of processes to develop secure systems. Several security requirements frameworks have been proposed but the benefits of these frameworks are limited to the realm of software. The most significant contribution of this thesis is to propose, apply and assess a security requirements engineering framework for cyber-physical systems that overcomes the issue of security requirements elicitation for cyber-physical systems. The proposed cyber-physical systems framework offers complete guidelines for practitioners and researchers to determine security requirements. A security requirements engineering Tool to facilitate application of our proposed framework has also been developed. The proposed framework has been evaluated by way of two case studies conducted on real-world cyber-physical systems implementations, which show promising results. Furthermore, this work also compares the activities mandated by our security requirements engineering framework with those of existing software security frameworks. The results of this thesis can be used as a basis for further research in security requirements engineering of cyber-physical systems. Organizations that apply the proposed framework derived from the results of this research will be better positioned to explore security requirements in the early phases of system development and be assured of an uncompromised system of security.

Acknowledgements

I would firstly like to thank God-Almighty for not only the ability, but also the constant strength and perseverance that kept me going through this thesis.

Further, I would like to express my deepest gratitude to my supervisor, Professor Volker Gruhn, who has been a steady guide through the course of my research and has always been supportive of my academic endeavours - no matter which corner of the Earth they led me to – thereby allowing me to interact with and learn from excellent researchers in my field from around the globe. I am thankful for dedicating his time and effort and for always making me feel welcome at his office. I would also like to express my appreciation for Professor Matthias Book, who not only offered deep critical insights through his review of my thesis, but also went out of his way to make available to me certain academic opportunities that I found to be extremely valuable.

I would also like to extend my thanks to all my colleagues in Paluno. It has been a great experience working with and exchanging ideas with them. All the talks, discussions and dissertation seminars have been invaluable experiences for me.

Last but not least, I would like to give thanks to my family, particularly my mother, who has always been a pillar of support for me, and without whom I would not be where I stand today. I thank her for her prayers, her emotional support and for always giving me advice when I hit a rough patch in life and don't know what to do. My wife too, has been a constant source of strength for me, a companion, a friend and both a mother and father for my children in my absence. I thank her for all of this and for not killing me when I told her I would be living in a different country for the next three years. I'd like to thank my daughter Eshal and my son Shaheer for bearing with me patiently and not holding against me the time I could not give them in pursuit of my doctoral studies.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

We are living in the era of digitization where software, system hardware, and sensors are working together with the aid of networks. This combination describes the concept of Cyber-Physical Systems (CPS) [1]. Modern societies and current economies heavily depend on advanced infrastructure for transportation, communication, energy and finance. Particularly these infrastructures rely on software systems and sensor networks. Threats in software systems or attacks on sensor networks cannot be afforded as there are important lives and organizational assets involved. In this situation, the urge to maintain security is of prime importance. Secure system development depends on an extensive focus on the process of requirements engineering for security. Software engineering gets its developmental supports from tools and techniques, as well as models that guide them to manage quality development support [2] [3]. These techniques provide information on how services are provided. However, when developing a secure system, one must consider the threats of the system as well.

Cyber-physical systems are the systems of systems that combine the physical world with the world of information processing. Cyber-physical systems include huge, complex systems like, power grids, management of transportation networks, smart cities, digital health-care, autonomous vehicles and telecommunications. Typically, they have large infrastructure and operate in a distributed environment. Most of these systems operate in a real time environment and are network connected for remote monitoring and controlling. This opens the way for an adversary to attack CPS components. Furthermore, CPS are gaining priority over other systems. The heterogeneity of these systems increases the importance of security. Both the developer and the requirement analyst must consider details of cyber-threats not only

for the software, but also the hardware, including both the sensor and the network [4] [5].

In present day software development industry, cyber-physical systems are gaining much attention from researchers and practitioners due to their high impact on the world's economy. These systems are considered as hallmarks of the modern age of computing power integrated with physical systems. With the rising use and importance of CPS, developers have come to terms with the importance of security in these systems, as any error - if left unhandled - can be fatal. For instance, any disturbance in the communication protocols of self-driving cars with minimal human intervention can be disastrous, leading to loss of not just the system itself, but also the lives that depend on its uncompromised satisfactory functionality.

Given that developing a secure system capable of safeguarding all interests of a client is not an easy task, it should come as no surprise that repeated attacks on unsecured systems have become quite commonplace, often resulting in losses of millions of dollars [6] [7]. Many software security breaches occur due to errors and misspecifications in analysis, design and implementation. Hence, emphasis on information security is gaining more and more importance in recent years. In this sense, security requirements engineering is an appropriate means to elucidate and determine security requirements at the analysis stage in Software Development Life Cycle (SDLC) [8].

Security Requirements Engineering (SRE) is the systematic process of eliciting, analysing, specifying and validating the security requirements of a system [9]. To develop a system with a security focus, a security requirements engineering framework is required. This is a set of guidelines which involve a sequence of activities to be used by researchers and developers to identify security requirements prior to implementation of the system. Several security requirements frameworks have been proposed. Among them, some of the famous ones are SQUARE, SREP, Secure Tropos,

CLASP, CORAS, and UMLsec [10] [11] [12] [13] [14]. The benefits of these frameworks are limited to the realm of software, and at some point, to supporting the computer hardware. Unfortunately, none of these frameworks focuses on addressing the new problem of cyber-physical systems, which result from the difference in architecture between classical and cyber-physical systems. The most prominent difference is the addition of the physical environment as an integral part of the CPS, necessitating a state of continuous communication with the rest of the system. In this regard, sensors to monitor the real world and much more extensive communication networks are of paramount significance. CPS communication involves a different form of data processing, of data incoming from the outer world and to be transmitted back to the outer world [15] [16]. As a result, CPS require a dedicated communicational channel for secure interaction. In addition, CPS have to meet real-time requirements as they control real-world processes, thus the risk from interference or break in communication becomes much greater than those for other systems. The diversity of cyber-physical systems forces the developer to take into consideration details of the security aspects of sensors, receivers, data processors, and communicators, as well as the general software security aspect, which are not addressed in most conventional security requirements frameworks.

Therefore, we propose a security requirements engineering framework for cyber-physical systems that overcomes the issue of security requirements elicitation for heterogeneous CPS components. The proposed framework supports the elicitation of security requirements while considering sensor, receiver protocol, network channel issues, along with software aspects.

1.1 Problem Statement

Cyber-physical systems integrate a various number of hardware, software and networking components with connections to the real physical environment. This and other unique characteristics of cyber-physical systems bear various opportunities and platforms for an attacker to launch an attack on the system.

Considering security and security requirements in the early stages of the development is an important step towards integrating appropriate security into the application and to protect the system from both any type of threat and its undesirable impacts. Although cyber-physical systems are an emerging field of research in recent years, a review of the latest literature did not reveal a thorough method for identifying security requirements for cyber-physical systems in the requirements engineering stage of the development process [17] [18]. Some degree of discussion can be found in literature pertaining to threats and vulnerabilities in the context of CPS, however the existing work does not contribute towards a comprehensive methodology for determining security requirements [19]. A major factor here is that most existing security requirements engineering frameworks were designed primarily with a focus on software-based systems, at a time when cyber-physical systems were still a relatively obscure concept.

In today's world, the software development industry is striving hard to increase productivity. Yet, this goal cannot divert the attention of the software development team from important aspects like security and risk assessment. Organizations from every industry and walk of life that utilize software-based systems have faced losses valued at billions of dollars due to major security attacks worldwide [6]. One of the major reasons behind the success of these attacks are incomplete and vague security requirements, often due to lack of attention to their elicitation and analysis [20].

Generally, security is considered as an afterthought to the software / system development, not realizing that security is not an afterthought but a very important aspect of the lifecycle. Extensive work by [21] [22] illustrates that if not considered in the preliminary phases of development, security issues can become hazardous for systems, particularly high risk systems used in military or autonomous vehicle systems. The incomplete security requirements may cause the product to be susceptible to failure as well as increase maintenance cost at later stages of the lifecycle. Consequently, security requirements are a key issue for cyber-physical systems. Security failure can lead to a host of problems, such as compromised confidential data that may result in a threat of physical harm to individuals or organizations. Stuxnet, Maroochy waste management systems and the water treatment plant are case-in-point examples of such CPS attacks. With the much heightened use of wireless network communication in CPS as compared to conventional systems, the threat to security becomes ever stronger.

If security requirements are not properly defined in the requirements engineering phase, then effective evaluation for success or failure of CPS components cannot be undertaken. Furthermore, there is no detailed security requirements engineering framework available for CPS. Thus, it becomes extremely important to address the security requirements in an early phase of software development life-cycle and to develop a comprehensive security requirements engineering framework for CPS.

1.2 Research Questions

The following are the major research questions that this work will address:

1. Which security threats are most important for cyber-physical systems?
2. What are the existing security requirements engineering frameworks to specify the security of software?
3. Do existing security requirements frameworks fulfil the needs of cyber-physical systems?
4. Which risk assessment technique can be utilized for the security requirements framework of cyber-physical systems?
5. How effective is the proposed SRE framework in eliciting of security requirements for cyber-physical systems?

1.3 Contribution

This thesis makes the following main contributions:

A systematic mapping study for cyber-physical systems is conducted with the following objectives:

i. To explore and consider security requirements engineering frameworks/ methods/techniques for software and cyber-physical systems proposed till date in literature.
ii. To understand security goals, threat and vulnerabilities identified in literature as essential for consideration in the security requirements engineering process.
iii. To investigate the methods of validation used for the security requirements solutions proposed in literature.

The study provides an overall view of the state-of-the-art frameworks / methods / techniques proposed till date to deal with security requirements. The results of this study provide insights to researchers and highlight the need for developing

frameworks to deal with security requirements for particular kinds of systems like cyber-physical systems. Also, it motivates future work to devise methods to cater to domain specific security risks and requirements.

In this thesis, our main contribution is to provide a comprehensive security requirements engineering (SRE) framework for cyber-physical systems that overcomes the issue of security requirements elicitation for CPS. The proposed CPS framework offers a set of procedures for practitioners and researchers to determine security requirements. The proposed CPS framework aims to serve as a complete guideline, through a number of activities, to analyze and identify threats as well as to determine security requirements of CPS while taking different aspects of CPS into account. The novelty of this work is that such an implementation with regards to problems of this scale has not been reported significantly in literature.

The proposed CPS framework is a systematic approach to incorporate security goals, threats, and risk assessment that are critical to the CPS. We have a set of 8 main activities, and one important technique called the "*misuse case*". The framework delineates the activities that are essential for requirement analysts to follow in order to identify the security requirements for CPS.

The CPS framework is in the form of a checklist that needs to be followed. The results (output) from each activity are fed to future activities. The framework proposes an agile methodology to select the required activity. The analyst has an array of activities to utilize, and may choose from them as fits his requirement, or otherwise adapt the framework to the current need.

The proposed CPS framework has been evaluated through case studies, in which we conducted two case studies. In the first case study, we applied our security requirements framework on a smart car parking system. A functional prototype for the demonstration of a smart car parking system was developed, consisting of a

physical and a software implementation. We analysed and identified the major security goals and threats of a cyber-physical systems. This analysis is based on a few matrices that were developed during the implementation of this case study. Application of the framework to the system led to elicitation of 43 major security requirements which were not immediately determinable otherwise. The second case study was conducted as an industrial case study at the Soccerwatch GmbH. The proposed CPS framework was applied to Soccerwatch's systems, which led to a similar scale of security requirements identification. The results were well appreciated by the Soccerwatch GmbH as these identified security requirements were found to be useful in furthering the interests of the organization. The results from these case studies support the case of the proposed framework and offer encouragement for more research in this direction.

Recently, cyber threats are on the rise, and in order to secure cyber-physical systems from being at risk, and hence minimize the economic and even life-threatening consequences, it has become increasingly imperative that a framework specially designed to cater to the needs of cyber-physical systems be devised. The increased demand for security in organizations also justifies the need for a systematic security requirements engineering framework, which would make organizations better positioned to explore security requirements in the early phases of software development and be assured of an uncompromised system of security. This research attempts to contribute towards this end by providing such a framework.

1.4 Publications

The figure 1.1 shows a list of journal and peer reviewed conference publications:

S.Nr	Publications	Related Chapters
1.	S. Rehman and V. Gruhn, An Effective Security Requirements Engineering Framework for Cyber-Physical Systems (2018), International Journal of Information and Communication Technologies, special issue Cyber-Physical Systems: Data Processing and Communication Architectures vol. 6, issue 3, p. 65 (ISSN: 2227-7080; ESCI-WoS index).	5, 6
2.	S. Rehman, M. Ceglia and V. Gruhn, Analysing Security Threats for Cyber-Physical Systems, Springer Future of Information and Communication Conference, (pp. 1095-1105) (Springer FICC), San Francisco-USA, March 2019.	2, 3
3.	S. Rehman, A. Iannella and V. Gruhn, A Security based Reference Architecture for Cyber-Physical Systems, Springer First International Conference on Applied Informatics, (pp. 157-169) (Springer ICAI 2018), Bogota-Colombia, November 2018.	2
4.	S. Rehman, V. Gruhn, S. Shafiq, I. Inayat, A Systematic Mapping Study of Security Requirements Engineering for Cyber-Physical Systems, Springer The 7th International Symposium on Security and Privacy on Internet of Things, (pp. 428-442) (Springer SpaCCS 2018), Melbourne-Australia, December 2018.	3, 4
5.	S. Rehman, C. Allgaier and V. Gruhn, Security Requirements Engineering: A Framework for Cyber-Physical Systems, IEEE 16th International Conference on Frontiers of Information Technology, (pp. 315-320) (IEEE FIT'18), Islamabad-Pakistan, December 2018.	5, 6
6.	S. Rehman and V. Gruhn, An Approach to Secure Smart Homes in Cyber-Physical Systems/Internet-of-Things (2018), The Fifth IEEE International Conference on Software Defined Systems, (pp. 126-129) (SDS-2018), Barcelona-Spain, April 2018.	6
7.	S. Rehman and V. Gruhn, Security Requirements Engineering (SRE) Framework for Cyber-Physical Systems (CPS): SRE for CPS (2017), New Trends in Intelligent Software Methodologies, Tools and Techniques: Proceedings of the 16th International Conference SoMeT_17 (vol. 297, pp. 153-163). IOS Press, Kitakyushu-Japan, September 2017.	4, 5
8.	S. Rehman and V. Gruhn, Recommended Architecture for Car Parking Management System based on Cyber-Physical Systems (2017), IEEE Engineering & MIS (ICEMIS), (pp. 1-6). IEEE, Monastir-Tunis, May 2017.	1, 6
9.	S. Rehman, M. Prehn and V. Gruhn, SmartChair: A Realization of Smart Health Care System based on Cyber-Physical Systems (2018), ACM. International Conference on Engineering & Management of Information System, (p. 20). ACM, Istanbul-Turkey, June 2018.	1
10.	S. Rehman, A. Hark and V. Gruhn, A Framework to handle Big Data for Cyber-Physical Systems (2017), The 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference, (pp. 72-78). IEEE, Vancouver-Canada, October 2019.	1, 2
11.	S. Rehman, M. Ceglia, S. Siddiqui and V. Gruhn, Towards an Importance of Security for Cyber-Physical Systems/Internet-of-Things, The 8th ACM International Conference on Software and Information Engineering (ICSIE 2019), (pp. 151-155). ACM, Cairo-Egypt, April 2019.	2, 5

Figure 1.1 Publications

1.5 Thesis Structure

In this chapter, we present the motivation for the work done as part of this thesis. The contributions of the research as well as peer reviewed conference and journal publications supporting the validity of the research is also presented. The remaining chapters will discuss the following:

Chapter 2 presents an outline of the security of cyber-physical systems where fundamental terms and definitions of security requirements engineering for CPS are described. The chapter discusses the correlation of terms such as security requirements engineering, security goals, threats and vulnerabilities of CPS. Furthermore, the attack points and potential attackers are described in this chapter.

Chapter 3 presents the systematic mapping study of security requirements engineering for cyber-physical systems. This chapter begins by discussing the background and related work, followed by a description of the research method, mapping design and evaluation techniques used in the literature.

Chapter 4 provides an overview of some of the most important security requirements frameworks. These include SQUARE, Microsoft SDL, UMLsec, Secure Tropos, CLASP, SREP and CORAS. The results of this comparison are presented where the strengths and weaknesses of each framework are provided. They are also analysed in the context of suitability to cyber-physical systems.

Chapter 5 proposes the security requirements engineering framework for cyber-physical systems. All activities of the proposed framework are explained. It also discusses how the activities of the framework apply to cyber-physical systems. The framework workflow process is described where input (to an activity), technique and output (from an activity) are described. The implementation of the framework through the use of a software tool developed by the author for this purpose is also illustrated in this chapter.

Chapter 6 evaluates the proposed framework by employing the two case studies of a smart car parking system and Soccerwatch. All activities of the framework are applied on these case studies to evaluate the effectiveness of the framework. The framework is used to determine the security requirements in each of these case studies successfully, and results are presented.

Chapter 7 provides the concluding remarks and discusses future work possibilities.

CHAPTER 2

Security of Cyber-Physical Systems

In this chapter, we discuss Cyber-Physical Systems (CPS) from a security perspective. We discuss how CPS security is inherently different from that of classical software and how that leads to security challenges, considering the emerging cyber-physical properties of most modern systems. We focus particularly on physical-security fundamentals in the context of CPS. We explain security requirements engineering and why there is a need for security requirements engineering for CPS. We provide examples where the interaction of functionality and diversified communication can lead to unexpected threats and vulnerabilities as well as produce larger impacts. Finally, we discuss main attack points of CPS, i.e., where an attacker can get access to CPS components easily.

2.1 Overview

Cyber-physical systems are used today to achieve unprecedented levels of functionality and process optimization across a wide range of industrial applications [23]. However, they have not been without their problems, particularly in the area of security, which can be defined as an attempt to protect the CPS components from malicious attacks, unauthorized access or damage to its physical parts. CPS have been the targets of some of the most widely known security breaches in recent history such as Stuxnet, Maroochy Waste Management System, etc [24] [25]. While security methods for both the cyber and the physical elements (manufacturer guidelines) of these systems are available, they alone do not seem to be able to solve the problem, as a result of the complex interdependencies and crossover effects involved, which naturally lead to unexpected threats and vulnerabilities [26]. Physical attacks can impairment or compromise the system and cyber-attacks may cause failure or malfunctioning the system. Because of the criticality of the application, any kind of

attack can lead to highly debilitating circumstances in the real world. As a result, it is imperative to determine the security requirements for CPS [27] [28].

Even though risk mitigation on the user end is also an important step in ensuring CPS security, our goal here is to enable future designers of CPS to be able to develop more secure, privacy-enhanced products with the help of incorporating security requirements in the requirements engineering phase.

2.2 Cyber-Physical Systems

Cyber-physical systems are the systems of systems that combine the physical world with the world of information processing. CPS involves interaction between heterogeneous components, that include electronic chips, software systems, sensors and actuators. As a result, a CPS environment is quite different from and more complex than conventional environments. This is particularly the case as CPS is designed to automatically adapt its strategy to the current environment in response to the monitored situation [29].

CPS are similar to Internet-of-Things (IoT) systems but feature greater coordination between physical and computational elements [30] [31]. The interaction between cyber-physical systems and their environment which consists of users, the physical environment and a variety of hardware and software-based systems are important features of CPS. This particularly involves integration, interoperation, monitoring and control of cyber-physical systems components. Unlike standalone devices, CPS have a chain of inputs and outputs associated with interacting elements [3] [15]. Furthermore, application of CPS cannot be narrowed down to any particular field, rather their applications extend to almost every field [32]. These systems will enable an advanced customization of health services, traffic management, finance, smart grid etc.

A CPS is characterized by an extreme variety of deployed technologies and a varying scale between such systems [33]. Technologies, such as computing devices, embedded systems, sensors, control units and other devices that serve distinct purposes, can be deployed in a CPS. For instance, one CPS might mainly consist of a few sensor and actuator nodes to monitor and adjust the room temperature. On the other hand, a CPS can grow to a structure of large heterogeneous and decentralized networks of distributed subsystems that could - for instance - perform different autonomous tasks on a solar energy plant [34]. In order to handle both this complexity and the changes in system scale, CPS feature adaptive capabilities. The complexity of a CPS is in most cases determined by the system's scale and the diversity of deployed components. Furthermore, most CPS employ advanced feedback control technologies. Feedback control refers to the ability to actuate cyber-physical events in response to sensed changes in phenomena from the physical environment [35].

CPS are usually composed of three layers: The physical layer, the application layer and the network layer as shown in figure 2.1.

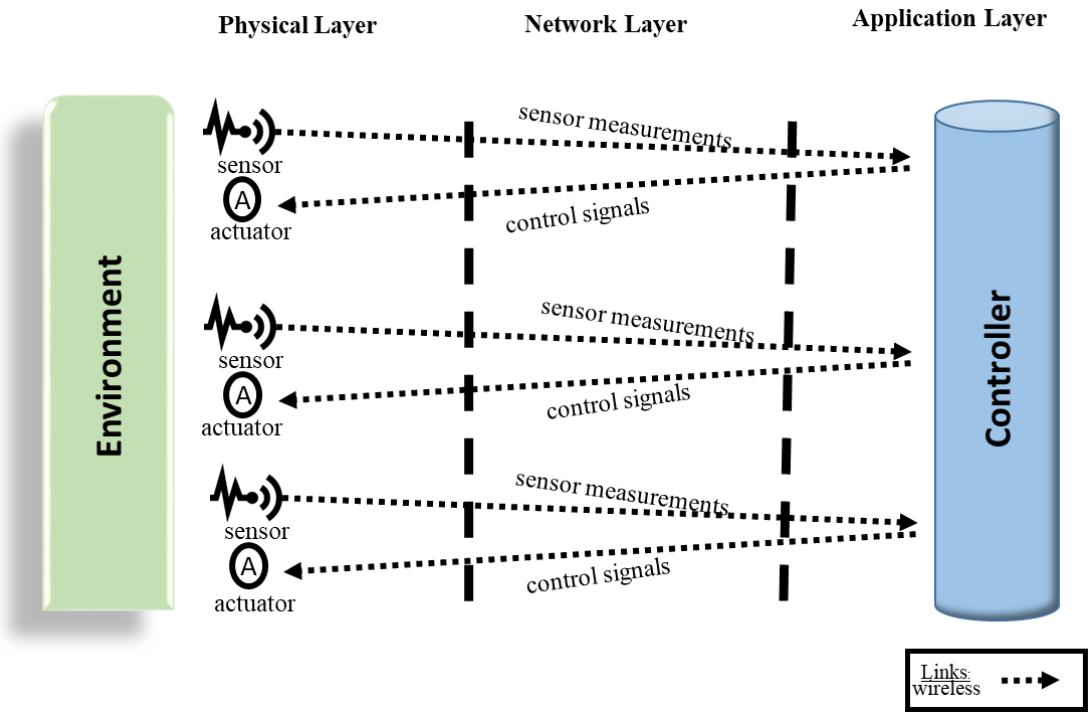

Figure 2.1 Architecture of Cyber-Physical Systems

2.2.1 The Physical layer

A physical layer is made up of sensors, actuators, a variety of other devices and subsystems with sensing, some degree of computing and communication capabilities [36]. Sensors in this layer record physical phenomena in the physical environment of the system. This layer also involves actuator units that respond to real-time monitoring of events as well as communicate with the application layer to enable data processing [37] [38]. The actuator units have capabilities to change the attributes of physical objects and phenomena in the physical environment. The aforementioned phenomena range from states of natural phenomena, such as the ambient temperature in an enclosed space, to man-made systems, such as in the case of a surgical room, up to complex combinations of both [39]. Actuation in a physical process is triggered by cyber-physical events that are generated based on the results of the information processed in the system. An important property of this layer is the ability of its components to communicate with external networks like the internet using a gateway node. Since this layer is particularly vulnerable to cyber-attacks, determining security requirements is essential.

2.2.2 The Network layer

The purpose of the network layer is to transmit control commands and sensory data between the application layer and the physical layer. The large number of different heterogeneous networks connected means that special security protocols have to be taken into account. The network layer is essential for CPS operation as it functions as a bridge between the physical and the application layers alongside being the intermediary between the sensors and actuators [40] [41]. The network layer is the communication channel for the exchanged data, measurements from the sensors and commands to the actuators. Some of the communication protocols used in the network layer are Ethernet, Distributed Network Protocol (DNP), Recommended Standard

(RS-232), Transmission Control Protocol/Internet Protocol (TCP/IP), dial-up modem, and other wireless protocols [42]. The most common mode for sensor and actuator communication is wireless, considering their distributed nature. For wireless communication, the cloud or a physical server is often used. The application layer and the physical layer work closely together in an interconnected manner via the network layer.

2.2.3 The Application layer

The application layer consists of various components like controllers, databases, and a form of user interface. This layer is the central part of CPS, which receives data from the network layer and produces control commands to controls the physical devices and processes [43]. The application layer is responsible for providing different services, particularly application-specific roles involving data reporting and representation, acting as a control panel for the end users, and offering a mostly graphical user interface for a range of applications and services, including user access, precision agriculture, environment monitoring, intelligent transportation, smart grid, smart home and more. It also provides the interface and services to the end users to access the sensory data using mobile devices or terminals, which may come in different forms, and so, they would have very specific security requirements.

2.3 Differences to Classical Systems

The communication of cyber-physical systems components between system and physical environment forms the basis for the fundamental difference from classical software, alongside the ability of CPS to interact with the real world through an extensive application of sensors / actuators / controllers which make it possible to monitor or to influence processes in the physical world [44].

Building on these fundamental differences, there are also naturally some specific requirements that differ from those of classical systems. For example, many CPS have

to meet real-time requirements because of the interaction with the physical world [45]. A system that controls the electrical power grid in an area is a good example of why real-time is so important for these systems. If an engine in a power plant malfunctions, the system must decide how to compensate the energy loss with other resources at its disposal. Otherwise the consequence would be a blackout, which might cause serious economic damages. To avoid this, the system must be able to solve such conflicts in a very short amount of time [46].

Cyber-physical systems are tightly integrated with physical processors and computing [47] . The complexity of CPS stems from the fact that these systems tend to employ far more operational elements than classical systems, particularly since classical systems only required a small number of such components [48]. The heterogeneity of different systems and devices that are combined in a CPS also describe a difference to classical systems [44]. CPS deal with a much larger variety of heterogeneous devices and systems. In the example of a traffic control system, the system could use the information that traffic cameras provide, but also the GPS data of smartphones or sensors that are used in cars and from many other sources. In this case, the cameras may work with different data formats, the smartphones could be from different providers using different operating systems and thus using many different data formats. The same is also true for car sensors or any other source of information that is directed to the CPS. The CPS should be able to work with these large volumes and types of different data and process it securely in real-time. This shows that how security for a CPS must deal with issues relating to each of the physical, network and application layers, differentiating it from security of classical systems.

2.4 Security Challenges

Cyber-physical systems are going through a revolutionary stage in their development, and therefore, face many challenges, security being one of the most important. Like classical software systems, CPS are prone to cyber-attacks which focus on obtaining internal data or interfering with data processing and storage [49]. Attackers hack into the system, spread malicious code or malware or aim at obtaining sensitive data which they can use for their malicious purposes such as threatening organizations or using the system under the guise of a legitimate user by stealing identity data [50]. Attacks on the 'cyber' part of a CPS disrupt the functionality that controls the cyber-physical events in a system [51].

Networks that are deployed in a CPS are threatened by a variety of network attacks that aim at intercepting and redirecting data flow on the communication links between components [52]. In terms of network communication, network protocols present a vulnerable point to the attacker. Communication protocols that provide insufficient or no security measures for network communication make it easy for attackers to target routing information and data flow between nodes. Attacks on networks do not only come from network communication participants that are controlled by attackers. An attacker could monitor or eavesdrop on the communication links between nodes and hence intercept information, e.g. by analyzing the communication [53] [18].

In addition to networks linking the router, control units and other components, network attacks also target sensor and actuator nodes that are organized in networks as well. Sensors and actuators are subject to strong security-critical constraints, i.e., their limited resources in terms of processing power, memory space and AC power, amongst others. Sensor and actuator networks are security-critical to the system as their resource constraints can be exploited with a level of effort most attackers are capable of. Captured sensor and actuator nodes then serve as entry points for further

attacks on the system, such as capturing secure communication keys or launching malicious code or malware [54]. Organizing sensors and actuators in wireless networks increases the risk for those networks as it can be assumed that wireless communication is insecure and provision of encryption mechanisms for deployed nodes are resource consuming operations. Insecure wireless network communication supports attempts by some attackers to deploy malicious nodes in a sensor or actuator network or capture a legitimate sensor or actuator node and overwrite its memory. By doing so, an attacker can mount wormholes or attract data traffic to sinkholes by updating unauthenticated routing information. In fact, sensors, actuators and the networks they are organized in are highly vulnerable to attacks. The entire network can be disabled by an attacker with sometimes as little effort as a single message to the right node [55].

Besides network attacks, sensor and actuator nodes face physical security challenges as well, given that they interact with the physical environment. An attacker could tamper with sensor data, for instance, by applying hot or cold objects to temperature sensing nodes. The measurement data that are transmitted to the system are thus misleading and might result in the system actuating undesired cyber-physical events. Sensors and actuators are exposed to the physical environment, which means that they are physically accessible. The service of these nodes can be disrupted by unintended accidents, indented sabotage, vandalism or even theft. Furthermore, Wireless Sensor Networks (WSN) that utilize cryptographic methods too face massive security challenges. Many of these methods are not feasible or heavily expensive due to computational constraints, power consumption, data sizes and processing times involved [56] [57] [58] [59] [60].

The security challenges described in this section provide an outline of typical issues CPS face when security measures are not sufficiently deployed, primarily a result of security requirements not being identified. The security risks of a system are

influenced by the system's purpose, its architecture, stakeholders and other factors. Hence security has to be considered in a CPS in response to its unique features.

2.5 Security Requirements Engineering

The aim of software security engineering is to properly address software security best practices, methodologies, processes, tools and techniques, in all stages of the software development life cycle [61]. The purpose of security is to protect the CPS from malicious attacks, unauthorized access or damage to the physical part.

Security requirements are defined as constraints on the functions of the system, and these constraints functionalize security goals, such as confidentiality, integrity, and availability [62]. The purpose of security requirements is to specify that the confidentiality, integrity, and availability of the software should be preserved. For example, the system shall not display customer's personal information to other organizations. Usually, these requirements specify the security goals that are required for the development of a secure system [22].

2.5.1 Why Security Requirements Engineering for CPS

Security Requirements Engineering (SRE) is an essential aspect of cyber-physical systems, but there is a lack of methodology defined to develop a secure software system. Though many methodologies and frameworks have been proposed for software, there is still a need to improve them [63]. Many researchers address the requirements engineering best practices and highlight the importance of system functionality, but a small amount of attention has been given to what the system should not do [64].

Different studies show that cyber threats have grown in the CPS environment, and there is a need to do more research to systematically handle security requirements [65] [66] [67] . Recently, many incidents of CPS attacks have been reported in the literature. In 2010, Stuxnet was the first documented cyber-attack on a CPS. It targeted a Siemens control system 'Supervisory Control and Data Acquisition (SCADA)' through

malware to control and destroy Iran's nuclear program. As a consequence, more than 50% of the Iranian nuclear infrastructure was attacked. Certainly, this incident creates an alarm for cyber threats [68]. In 2000, an Australian man was found guilty when he attacked the Maroochy waste management systems and released one million liters of impure sewage into rivers and local parks [69]. In 2006, a hacker infiltrated a US water filtration plant with malware that changed the levels of chemicals being used to treat tap water, and thousands of homes were affected in Illinois, USA [70]. Other famous cyber-attacks are Duqu and Flame, which were used to gain unauthorized access to their respective target systems [71] [72].

The problem of inadequately determined security requirements is not only faced by cyber-physical systems. In fact, software systems - that tend to be not as complex as CPS - have also been faced with a similar range of challenges over the past several years. The presence of these challenges presents a powerful argument towards the need for systematic SRE frameworks for such systems. For instance, it is estimated that the software development budget to fix security flaws is almost 75% of the total cost after handover of the product to the customer. This is an enormous amount of spending that builds mistrust amongst customers [73].

Software security engineering proposes many tools, techniques, methods, and best practices to develop a secure system [74] [75]. There has historically been a lack of understanding of software security that is essential knowledge relating to elements that need to be clarified and managed in the early phase of the SDLC [76] [77]. Therefore, developers have been relatively unsuccessful in implementing a secure software system when applying software engineering best practices [63]. For many systems, the security of software is not considered at the very beginning of the SDLC; it is only incorporated in the later stages of software development [78]. However, the significance of addressing security from the very beginning of system development has now become widely accepted in the research community [79]. It has also become apparent how there are increased risks of security threats that are introduced in

various stages of software development [80]. In light of this, integrating security requirements right at the beginning not only ensures secure software, but also saves precious time and reduces the effort of reworking for the software development team. In order to support the process of determining security requirements at the initial stage, we need a security requirements framework for CPS.

2.5.2 Security Issues around Sensor Networks

Since many cyber-physical systems depend on sensor networks, their security is an important factor to consider, as a malicious attack could harm or damage the physical part of the system. The application areas of sensor networks are very wide. They range from monitoring machines in production to military applications. The data are processed in networks, which makes their security critical. This is also due to the fact that sensor networks have new security requirements which are not matched by the security techniques of traditional networks [81]. One reason for this is that the sensors are partly located in open, accessible areas. This make them more vulnerable, and they translate to potential attacks. This is an important aspect of security of sensor networks that should be considered for every component [82]. If this is not done, unprotected components are vulnerable to attacks.

Confidentiality is also a subject of major importance within sensor networks. Networks could be used, in the worst case, to spy on individuals [83]. An example would be the long-term monitoring of persons or vehicles on a routine basis. Another aspect that affects the security of sensor networks are attacks on communication between the physical environment and the gateway to controller/server. In the simplest form, the attacker sends a high-energy signal to the sensor to prevent communication within the physical layer to the network layer. This can lead to serious consequences, especially in the case of security-critical CPS. Military applications are also threatened by such attacks. A possibility to combat these attacks lies in the nature of the networks themselves. If a part of the network has been compromised, this part can be demarcated and the communication can be routed around it [84] [85].

Therefore, it is of utmost importance to address the security of all three layers (i.e., physical layer, network layer and application layer) of CPS.

Given this wide array of possibilities for CPS security to be compromised, the need for ensuring security becomes all the more critical. We have seen that the greatest vulnerability that is specific to CPS lies in the domain of the perception layer, i.e., the sensor network. It naturally follows that this should be the primary area of focus for CPS security efforts.

2.6 Security Goals for a System

The acceptance of cyber-physical systems in society depends on trust from users that must be earned. This trust can only be gained by providing adequate security goals to users. Security goals aim to protect the system from threats and vulnerabilities and reduce risk factors. We aim to extend our understanding of security goals for CPS. For instance, in the case of sensor data oriented systems with multiple sensor nodes generating data, security is crucial to be sure that the data generated is coming from a trustworthy source. We can see how data authentication and other similar ideals can be very important security goals in CPS. The following is a list of some of the more common and important security goals:

2.6.1 Authentication

Authentication and accordingly, authorization concerns processes such as sensing, network communication and actuation. It needs to be ensured that data, transactions and communication channels can be trusted. Nodes (sensors) should be identified and authenticated before adding them to the network. Authentication in a CPS is considered difficult to achieve as, in some cases, it requires heterogeneous network authentication. The lack of an authentication process can lead to exposure of the network/information to an unauthorized user [86].

2.6.2 Availability

Availability of cyber-physical systems aims at protecting services such as processing and storing of information, communication or control of the physical process from any corruptions due to hardware failures, power outages or Denial of Service (DoS) attacks. Availability ensures that the information is available all the time to the authorized user when required. The possibility of risk increases when there is a DoS attack or service distractions as a result of a hardware failure, systems updates, or power failure [87].

2.6.3 Integrity

Integrity denotes that the information is accurate and trustworthy to the users. Integrity is disrupted when an information is modified in an unauthorized manner. Integrity in a system ensures that the data cannot be modified in any manner. Integrity of cyber-physical systems refers to the protection of information sent and received by sensors, actuators and controllers from so called deception attacks. Deception attacks aim at modifying transmitted information in an unauthorized manner and in a way, that makes the receiver believe that information is correct and unmodified [87] [88].

2.6.4 Confidentiality

The data being transferred within the network is inaccessible to an unauthorized user. The risks associated with confidentiality involve exposure of network information to an unauthorized user. Confidentiality of cyber-physical systems is an important factor for enabling user privacy when sensitive data are transmitted between and stored in different components of the system. In order to ensure confidentiality, the CPS needs to integrate security requirements against attacks like eavesdropping on the communication channels between control units and sensors/ actuators [89].

2.7 Threats of Cyber-Physical Systems

Threat could be anything that may harm the cyber-physical systems and it refers to potential dangers that can compromise security and bring harm to the system like cyber-attacks or failures. If no security requirements are implemented to defend against threats in the system, then the system becomes easy prey to corruption by attackers. Unauthorized access to the system may also lead to massive losses if attackers are able to retrieve critical information. Below, some of the more common security threats are detailed on physical, network and application layer.

2.7.1 Threats on Physical Layer

The threats faced by this layer revolve around the way the system interacts with the physical world and the physical devices in its domain. These devices may come under threat of physical damage or compromised and / or manipulated sensory or actuator signals. This breach of integrity in the physical layer can have critical, even catastrophic results, as faulty input data results in non-ideal control decisions that may lead to entirely unpredicted effects in the physical realm. These attacks can be conducted by accessing and manipulating internal nodes of the network, or employing external nodes to obtain system information or incapacitate the system operation [90].

2.7.1.1 Physical Attack and Natural Disaster

The physical parts of cyber-physical systems may be deployed to averse environments where they are exposed to access by external actors that may deliberately or inadvertently (natural disasters) interrupt CPS operation [91]. A single system element being compromised may lead to other successive dependent elements to also be affected, resulting in the entire system being under threat. Irrespective of whether the damage done to the physical layer was intentional or otherwise, physical damage in many cases is the most critical. This is because it is not possible to remotely rectify the problem over the network, and on-site presence of, in most cases, experienced

human personnel is necessary for damage rectification. Furthermore, the attacker could harm physical devices such as hardware, sensors, cameras, terminals etc. Such attack could threaten human lives and this must be prevented at any cost. If these systems are attacked from outside, this could cause great harm. Natural disasters can also lead to a loss of human lives, and the sensors or actuators would be unusable, and the financial damages would be significantly high [92].

2.7.1.2 Radio Frequency Jamming

Radio Frequency (RF) jamming aims to paralyze communication from the physical environment. This may interfere with the interaction of sensors to the controller or any gateways. Usually, the radio frequency jamming occurs with radio signals to detach the sensor communication tag through electromagnetic waves or high-level traffic of signals [93].

2.7.1.3 Sensor Node Compromising

Sensor node compromising is an active attack against a sensor node. By physically accessing a node in the network, an attacker gains control of it and the whole system is under threat [94]. The attacker could block flow of data or even determine the cryptographic keys that the node uses. Using these keys the attacker could also be able to compute additional keys or decrypt data [95] . For instance, an attack on the sensor measurements in an unstable critical physical process can result in major damage to the system. However, more critical scenarios are also conceivable [96].

2.7.1.4 Node Replication Attack

This form of attack involves a hostile actor introducing a new node to the network infrastructure. This node could function in several different roles. This includes actively corrupting or rerouting data packets, impersonating an authorized node to obtain, process, send, redirect or stop data transfer to and from a part of the physical layer, falsifying data (spoofing attacks), sending malicious command signals or merely eavesdropping into the network to extract cryptographic keys [97].

2.7.2 Threats on Network Layer

The network is host to the most critical information in its role as the channel of communication for the cyber-physical systems. This makes it a very likely target of what is termed a cyber-attack. The attack can be aimed at tracing confidential personal or organizational information or to discover important system related technical data that may be used to cause harm to the system. Cyber-attacks may also take a more active form by interfering in the routing procedure of data packets or altering the content of the data itself [98].

2.7.2.1 Denial-of-Service Attack

In a Denial-of-Service (DoS) attack the network of the cyber-physical systems is flooded with data or constantly receives invalid data. This leads the system to a state in which the traffic cannot be processed correctly anymore and the network service breaks down and the normal operation of the system cannot be continued. Due to the strong real-time constraints in a CPS, the system is especially vulnerable to DoS attacks. Unlike traditional information applications that may operate normally when the system becomes available again, a DoS attack on a CPS strongly compromises its availability [99].

2.7.2.2 Eavesdropping

There are several different ways to attack cyber-physical systems. One common threat is the leakage of data or eavesdropping, which is used to intercept the exchanged data [100]. As a result of intercepting sensitive information between sensor nodes, an attacker is able to violate user and owner privacy by monitoring the communication between those nodes, causing harm by theft of sensitive information, particularly in a medical or smart home environment. This type of attack is termed a passive attack, since the attacker does not directly interfere with the system and change its behavior or corrupt it. The attacker passively monitors the information that is sent being exchanged, silently compromising confidentiality and privacy. In a military context,

interception of tactical information is of particularly high relevance. In the case of industrial espionage, this can lead to serious damage to the organization or benefit to its competitors [101] [102].

2.7.2.3 Compromised-Key Attack or Data Tampering

In a compromised-key attack, the attacker gains access to a key for the system, and thus, can modify data. The attacker can also access other areas of the system. This is done without the actual users of the system being aware of it [103]. Data tampering means that legitimate data is intercepted and then modified by an attacker. This data is then sent to the original recipient. This fake or flawed data can induce abnormal behavior in the system because computations and responses are then based on inaccurate or completely erroneous measurements. Also, users will be receiving a flow of false information. Thus, the system authenticity becomes compromised [104].

2.7.2.4 Man-In-the-Middle Attack

Man-in-the-middle attacks are attacks that involve sending incorrect information. If it is not recognized as false, it influences the function of the system. In the case of a system which controls the operation of train switches, such attacks can lead to malfunctions or collisions of trains. An attacker aims to let certain nodes trigger actions which lead to undesired events in the system or prevent the nodes from taking actions against undesired events. To do so, the attacker fabricates certain messages and transmits them to these nodes [105].

2.7.2.5 Wormhole Attack

A wormhole attack is one in which the attacker uses a malicious node to redirect data received to another location in the network. This can be performed without compromising a node in the network. In a WSN, this attack could be performed at the initial stage when the sensor nodes communicate to find their neighbors. The attacker just replays a node's routing request to its neighbors. Since these new nodes now assume they are within range of the original node that initiated communication, a

wormhole has been created. With this attack the attacker is able to disrupt the routing, analyze the traffic and manipulate the sequence of packets [106].

2.7.3 Threats on Application Layer

The main target in this layer is the control unit, which tends to be less of a vulnerability than the other layers. Nevertheless, the proper working of the cyber-physical systems depends just as much on the timeliness of the application layer as it depends on other layers. Since the application layer also forms the interface between the system and the user, it becomes vulnerable to (mostly) unintentional human errors, which may result in incorrect actions taken by the system. The application layer in many cases may be susceptible to large-scale attacks on the user-interface intended to paralyze the system [107].

2.7.3.1 Malicious Software

Malicious software like Viruses, Trojans and Worms can harm the system in different ways by affecting privacy, confidentiality, integrity and availability of the system [108]. Malicious software is not a specific threat to CPS but to all systems. One example is Stuxnet. Stuxnet corrupted multiple computers by using a zero-day exploit in the software, making it difficult to counter these types of attacks.

2.7.3.2 Unauthorized Access

Unauthorized access to the data is a real threat that should be handled at the beginning of SDLC. It is quite possible that an attacker can easily access user information. This information can be injected through network communication or sensor nodes. The system has to make sure that an attacker is not able to illegally access data and resources on the system. If a person is able to illegally access data then confidentiality and data integrity would be harmed [109].

2.7.3.3 Social Engineering

In social engineering, the attacker develops the relationship with user and try to obtained the user confidential information. This could be information of various kinds, but most recurrently is found to involve passwords or bank information, or access personal or organizational workstations to secretly install malicious software that can give them access to the information mentioned above, as well as give them control over that computer or entry into the organization's network [110].

2.7.3.4 Data Manipulation / Tampering

Data manipulation or tampering is a common threat in application layer. Data can be tampered in different ways. An attacker incorrect data or modify the data into the fields of database. An attacker is able to just steal information of the database itself such as version and type. They can also be used to steal entries in the database like passwords, usernames, system data and sensitive user or organization data. Other intentions are to modify, delete or add data so that it could be possible for the attacker to legitimately access the system [111]. This threat harms the integrity of data.

Looking at these threats and their consequences, it becomes clear that security for cyber-physical systems have a central role to play in the development of these systems. If this aspect is not taken into account, attackers are free to access data and abuse systems as they wish. In fact, dangers would arise which are hardly imaginable. As a result, the value and functionality offered by cyber-physical systems would be almost completely lost.

2.8 Vulnerabilities in Cyber-Physical Systems

Vulnerabilities are another security challenge in cyber-physical systems next to threats. Vulnerabilities refer to weaknesses in systems or protocols and allow an attacker to perform attacks against the system. Vulnerabilities derive from defects, misconfigurations and network mismanagement [112]. Vulnerabilities in software systems also often originate from programming errors. According to Zeng [103]

hackers try to find vulnerabilities in existing software in order to create malicious software to harm the system. In CPS, the vulnerability may be due to defects on the platform, incorrectly configured or poorly maintained operating systems or hardware devices [19].

Table 2.1 Vulnerabilities in cyber-physical systems

Application	**Network**	**Physical**
No strong password policy	Password not encrypted during transmission	Unauthorized personnel can access devices
Antivirus and Malware protection is not up to date	Firewall does not exist	Poor configuration of hardware devices
Running unnecessary services	Certification between client and server is insufficient	Low quality of sensor/gateways
Operating system and software patches are not up to date or not maintained	Data protection between client and access point is inadequate	No proper hardware installation/vendors are unknown
Unregistered software	Network hardware	No exact system operational stability or Disaster Recovery Plan (DRP)
Malware Protection is not installed	Network parameter	Frequent updates or configuration
Intrusion detection/prevention software is not installed	Network communication	No physical protection for sensor/actuators
Logs are not maintained	Connectivity	Unregistered hardware or vendors are unknown
		No standby power/devices

2.9 Attack points in Cyber-Physical Systems

In order to achieve a secure CPS, security must be integrated into each component, since components that have been developed without taking security requirements into account can become a point of attack.

The attack points in figure 2.2 provide some examples of the different security threats that an arbitrary cyber-physical system might face. It can be seen from the figure 2.2 that threats can be directed at any component and every communication line in the system. It also makes clear that security of both each individual part as well as of their integration needs not only to be considered but also strictly planned and executed. The figure shows the attack points in various components of CPS. Each of these components has its custom attack points and should be secured in an appropriate way.

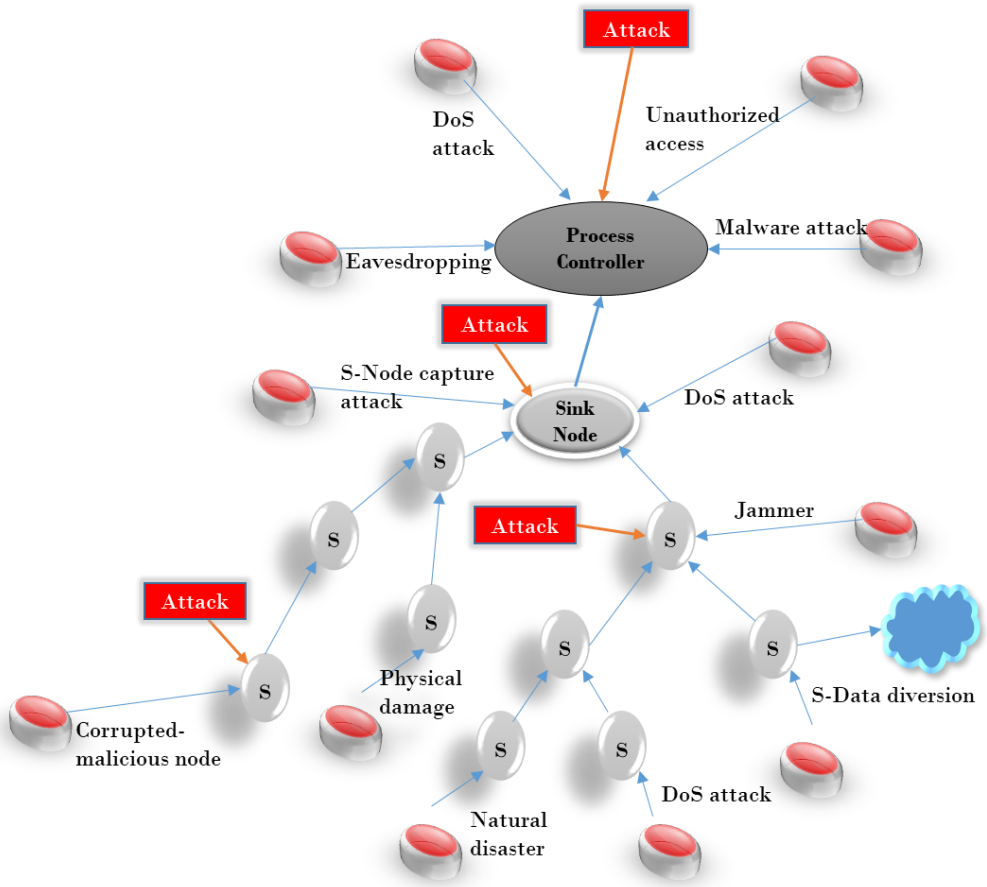

Figure 2.2 Attack points in cyber-physical systems

Individual sensor nodes in a WSN are inherently resource constrained [113]. This comes as a result of economic viability considerations, leading to sensor devices being limited in their power, computation, and communication capabilities. Added to that is the fact that sensor nodes are often deployed in accessible areas, making them more vulnerable to physical attack. Also, sensor networks interact closely with their physical environments and with people, posing new security problems [114]. These unique challenges in WSN tell that the new security requirements in WSN are needed.

2.10 Potential Attackers

Potential threats and attacks relate to different types of attackers, with unique intentions, motivations and capabilities. Analysis of potential attacker types that could harm the system can help to adopt and implement appropriate policies and strategies to prevent or mitigate the impact of an attack [115].

Attackers can be categorized in four main types: cybercriminals or skilled hackers, disgruntled insiders, criminal groups and nation states. The first category, cybercriminals, use their abilities and capabilities to find and exploit vulnerabilities in the system to compromise computers or devices. In contrast to cybercriminals who obviously threaten a system, Cárdenas et al. [116] point out that disgruntled insiders are a major security risk. Insiders, such as employees or business partners, have access to computers, networks and devices in the system, even though the access from outside is restricted. Moreover, their knowledge of the target system is often sufficient to gain access to sensitive areas and to cause damage to the system or its data. Criminal groups aim to attack a system for extortion. Nation states might use their capabilities to, amongst others, attack critical cyber-physical facilities of other nations [117].

2.11 Chapter Summary

Security for cyber-physical systems is very crucial due to the nature of CPS and their interaction with the physical world. In this chapter, we presented an introduction of security for cyber-physical systems and security challenges. Cyber-physical systems have additional security requirements in comparison to classical systems. This results from the inclusion of sensor networks and actuators which are also referred to as the perception layer. The perception layer consists of low cost devices that have limited computation power and small storage. Since they are deployed in unattended environments, their security differs from classical systems. Attacker can easily access these devices, making them prone to being compromised. Therefore, we illustrated the differing needs for security of CPS, distinguishing between classical software and

CPS. We discussed security requirements engineering and need of security requirements for CPS. Furthermore, we analyzed security goals and threats for CPS, with particular attention to each of the physical, network and application layers respectively. We described the main attack points on CPS and discussed the potential attackers for CPS.

CHAPTER 3

Systematic Mapping Study of Security Requirements Engineering

Since the world is moving towards secure systems, security has become a primary concern and not an afterthought in software development. Secure software development involves security at each step of the development lifecycle from requirements phase to testing. With a surging focus on security requirements, we can see an increase in frameworks / methods / techniques proposed to deal with security requirements for variable applications.

However, to summarise the literature findings till date and to propose further ways to handle security requirements, a systematic and comprehensive review is needed. Our objective is to conduct a systematic mapping study:

(i) to explore and investigate security requirements engineering frameworks/methods/techniques proposed till date.

(ii) to determine the security threats and security goals reported in literature.

We conducted a systematic mapping study for which we defined our goals and determined research questions. We then defined exclusion criteria and designed the map systematically based on the research questions. The search yielded 337 articles after deploying the query on multiple databases and refining the search iteratively through a multistep process. The mapping study identified and categorised the existing requirement engineering frameworks / methods / techniques proposed to deal with security requirements for multiple domains and also focused on their implementation and evaluation mechanisms. Second, we identified and categorised the security requirements and threats reported in the selected studies. The study provides an overall view of the state-of-the-art frameworks / methods / techniques proposed till date to deal with security requirements. The results of this study provide

insight to researchers, particularly with respect to the importance of focusing more on developing frameworks to deal with security requirements for specific kinds of systems like cyber-physical systems. Also, it motivates future work to devise methods to cater to domain specific security risks and requirements. Furthermore, it provides directions to where the future research is heading in this domain.

The rest of the chapter is organized as follows: Section 3.1 provides the introduction. Section 3.2 describes the background and related work in the area. Section 3.3 describes the research methodology and the selection process for the articles. Section 3.4 and 3.5 explain the construction of a systematic map of the study and its mapping design. Article evaluation is described in section 3.6 and conclusion of the study is described in section 3.7.

In this chapter, we have taken a brief look at the main frameworks or approaches that are proposed in the literature, mentioning their salient features. In Chapter 4, we choose the most important of these frameworks and describe them in depth, as well as performing a detailed comparison of their constituent activities.

During the course of this systematic mapping study, we identified the common threats faced by CPS reported in literature, and the desirable security goals. This list of threats and security goals was then used to form a basis for Chapter 2 of this work, where a more detailed discussion on threats and security goals in the context of CPS was presented.

The findings in this chapter concerning previously proposed frameworks which we detailed in Chapter 4, particularly with regards to the activities included in each of these frameworks, gave substantial support to formulating our own framework proposed in Chapter 5. Also, the evaluation methods for these frameworks found in this systematic mapping study motivated the use of case-study based evaluation for the proposed framework which is presented in Chapter 6.

3.1 Introduction

In today's world, the software development industry is striving hard to increase productivity. Yet this goal cannot divert the software development team's attention from important aspects like security and risk assessment [118]. Software-based industry has faced losses valued at billions of dollars due to major security attacks worldwide [6]. One of the major reasons behind these attacks are incomplete and vague requirement elicitation and analysis [20]. Professionals seem to have developed an interest in security requirements engineering and have started considering it as a preliminary step towards secure and efficient software development.

In present day software development industry, cyber-physical systems are gaining much attention of the researchers and practitioners due to their high impact on the world's economy. These systems are the bridge to the modern age of computing power with support for physical systems [119]. With the growing importance and use of CPS, developers have come to terms with the importance of security in these systems, given that any error if left unhandled has proven to be potentially fatal. For instance, any disturbance in the communication protocols of self-driving cars with minimal human intervention can lead to a disastrous collision.

To summarize, the importance of security is well-established in academic circles for modern day systems including CPS. Software development teams need to know that security is not an afterthought but a very important aspect of the lifecycle and that it has been shown that if not considered in the preliminary phases of development, security issues can become hazardous for systems, particularly safety critical systems used in the health or defense industries. Therefore, the main aim of the study is to identify the security requirements engineering solutions that help in ensuring maximum security and also in understanding the threats, vulnerabilities, and security requirements of these systems. The study provides an overview of the current techniques on SRE available in literature and motivates the practitioners and researchers to strengthen this area of research by providing implications.

3.2 Background and Related Work

Security Requirements Engineering (SRE) has evolved over recent years and an increasing number of security frameworks have been proposed in the research community. Now there are many different approaches for security requirements engineering, including multilateral (SQUARE), UML-based (UMLsec), goal-oriented (KAOS/Tropos) and Common Criteria-based approaches (SREP). Multilateral approaches are more up-to-date than unilateral approaches because they take into consideration and attempt to reach a compromise between the views of different stakeholders, which is an important and integral part of the process [120]. There are a number of proposals that attempt to address security concerns early in the development lifecycle which are not specific to CPS. There have been several studies till date, available in literature, that focused on reviewing security requirements engineering from various perspectives as discussed below:

Mellado et al. [121] studied the software requirements engineering techniques proposed in the information systems (IS) literature. The study revealed interesting insights regarding the techniques, the models and their integration of standards. The paper summarized the existing techniques to provide the researchers with an overview of which particular techniques are suitable in certain respective implementations. In another paper, Gopal et al. [122] describe that the security requirements play an important role in system development. There are a number of security requirements methodologies that have been developed. Researchers are still working on enhancing current methodologies or developing new ones to make a system secure. Risk assessment, asset management, validation of functional and non-functional requirements and security requirements elicitation are significant parts of a security requirements method. The most famous security requirement methods in use today are SQUARE, CORAS, UMLSec, and Secure Tropos, though each fails to perform one or more of these functions adequately [14] [123] [124] [125].

Yahya et al. [126] conducted a review on tool supports for security requirements engineering. The study has evaluated seven tools developed to deal with security requirements engineering and identified the problems and gaps existing. The study concluded that a considerable amount of research has been conducted on formalizing a model to capture the security requirements. However, capturing requirements from textual representations still needs attention. Therefore, the authors plan to explore Essential Use Case (EUC) approach in order to deal with the problem.

Tondel at al. [127] point out that the most software developers are not interested in security requirements, primarily due to a lack of knowledge concerning security requirements engineering. The greater portion of the interest lies in implementing the functionality of the software and this leads to negligence with regards to the security requirements. However, due to an increase in security awareness and demand, developers now realize the importance of security and have turned their focus to also include security requirements. Therefore, the authors compare some security requirements framework and identify security requirements which are most important in order to incorporate them at the beginning of software development. They concluded from the comparison that the most important security features are security objectives, assets and threats.

Yoo and Shon [128] examine the different security standards and protocols for network communications and identify security vulnerabilities and security requirements for cyber-physical systems. The details of IEC 61850 [129], an international standard for network communication is presented for CPS as cyber threats have increased in the CPS environment and researchers expose the vulnerabilities and security requirements for separate protocols. To handle the security vulnerabilities and security requirements, the architecture of security requirements is proposed. The architecture has six security layers that include: network security, protocol security, gateway system security, security service

mapping, configuration tool security, and proper protocol mapping, each layer describes the security requirements separately.

Beckers et al. [130] made a comparison of different security requirements engineering methodologies. They have defined criteria based on literature and also on risk analysis defined by ISO standards [131]. Upon these criteria they evaluate various SRE methodologies. For example, one parameter of these criteria is to analyze which SRE methodology should be used to semi-automatically formulate from security requirements at an early stage of SDLC and the other parameter concerns which SRE methodology should be used to examine the security protection against attacks. Based on these criteria, they concluded that KAOS [132] and secure i* [133] are well-suited SRE methodologies as they are using model / standard of development and they are validated formally. However, these methodologies do not cover all aspects of risk analysis but the authors claim that extending these methodologies is more feasible.

Salini and Kanmani [9] provide a short introduction of different security requirements engineering methods and compare them. This describes the benefit of being able to differentiate between security requirements and to define properly the mechanisms to achieve security. It also offers good examples of different types of security requirements methods. The paper also evaluated the main activities of security requirements engineering and provides the procedure of security requirements in later phases of SDLC. The analyst team can easily adopt any of these SRE methods according to their needs and expectations.

It can be inferred from the existing literature that though there exist a number of review papers on security requirements engineering in general, including reviews and comparative analyses of existing frameworks, no such effort has been done in the context of CPS. Thus, there is a need of a comprehensive review of security requirements for cyber-physical systems, techniques / frameworks / tools / techniques in use to handle security requirements and to identify the threats and vulnerabilities.

3.3 Research Method

3.3.1 Goals

Following are the goals of this systematic mapping study:

Goal-1: To identify the existing security requirements engineering solutions proposed for software and cyber-physical systems in literature.

Goal-2: To understand the security goals, threats, and vulnerabilities that are essential for a requirement analyst to consider in order to identify the security requirements for CPS.

Goal-3: To investigate the process of validation for the existing security requirements engineering solutions.

3.3.2 Research Questions

The goals of the study are then refined into the formulated research questions.

RQ 1: Which security requirements engineering solutions for software and cyber-physical systems exist in literature?

RQ 2: How to implement these security requirements engineering solutions?

RQ 3.1: Which security goals are considered important particularly for CPS?

RQ 3.2: What are the main security threats and vulnerabilities for CPS?

RQ 4: What empirical methods are used to evaluate the proposed security requirements engineering solutions?

3.3.3 Articles Selection Process

3.3.3.1 Search String

The search strategy is designed to obtain maximum articles relevant to the area and scope of the study. For this purpose, The query string is formulated based on the guidelines of Petersen et al. [134]. The query string shown below is applied on four well known database repositories which include IEEE Xplore, ACM Digital Library, Springer and Elsevier.

((Security requirements) OR (Security requirements engineering) OR (Security requirements engineering methodology) OR (Security requirements engineering process) OR (Security requirements engineering framework) OR (Security requirements engineering for cyber-physical systems) OR (cyber-physical systems))

The query yielded highly relevant articles, contributing an initial pool of 337 articles in total. The obtained articles were assessed based on our quality assessment process as shown in table 3.1. We assessed each article accordingly and annotated and categorized it "Accepted" and "Rejected".

3.3.3.2 Exclusion Criteria

Following is the criteria defined for the exclusion of articles.

EC-1: Articles that are not written in English are excluded

EC-2: Articles that are not peer reviewed are excluded

EC-3: Articles that belong to magazines or non-peer reviewed venues are excluded

EC-4: Articles that are not relevant to the scope of the study are excluded

EC-5: Articles of length less than 4 pages are excluded

The articles then underwent the scrutiny process through our defined exclusion criteria. The article selection with reference to the processes involved in the scrutiny is summarized in table 3.1.

Table 3.1 Quality Assessment Process

Phase	Method	Assessment Criteria	Count
1st Phase	Identify articles using search string	Keywords/ Query string execution	337
2nd Phase	Remove duplicate articles	Duplicate removal	336
3rd Phase	Exclude articles based on titles	Search strings in titles "Accepted" "Rejected"	322
4th Phase	Exclude articles based on abstracts	Search strings in abstracts "Accepted" "Rejected"	317
5th Phase	Obtain selected articles that pertain to the goals of the study	Addressing security requirement engineering and cyber physical systems, empirical studies "Accepted" "Rejected"	313

Details from each individual repository and the final pool is shown in table 3.2 which turned out to be of 313 articles after the successful assessment of the highly relevant articles.

Table 3.2 Number of Articles obtained from each individual repository

Repositories	Relevant articles obtained	Final Pool (5th Phase)
IEEE	215	195
ACM DL	54	51
Springer	24	24
Elsevier	44	43
Total	337	313

3.4. Research Protocol

In order to make the study meaningful and systematic, we employed a research protocol. The research protocol is followed based on the guidelines of Petersen et al [134]. The protocol establishes the criteria to be followed while the study is being carried out. The protocol that we have followed in the study is shown in figure 3.1.

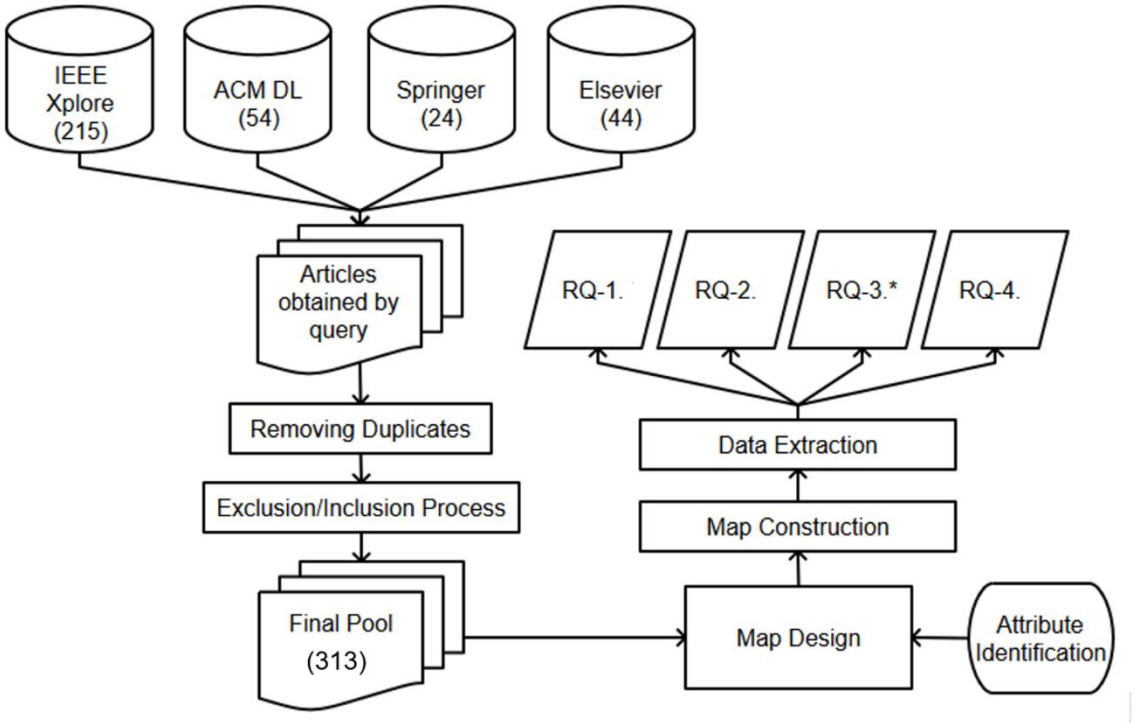

Figure 3.1 Research Protocol

3.5. Mapping Design

3.5.1. Research Map

Initially, we constructed a research map based on the above research protocol. The map establishes the baseline of the study on which the evaluation of the papers is done. The identified articles were then categorized based on their evaluation results. Formulated research questions and the identified attributes along with their description is shown in table 3.3.

Table 3.3 Research Map

Research Questions	Attributes	Description
RQ-1	Security requirement Solutions	To identify security requirement engineering frameworks, tools and techniques proposed in the articles
RQ-2	Implement of Security requirement Solutions	To identify the implementation of security requirement engineering frameworks, tools and techniques proposed in the articles
RQ-3*	Security goals, Security threats, vulnerabilities	To identify security goals, threats and vulnerabilities mentioned explicitly in the articles
RQ-4	Evaluation method	To identify how the articles are evaluated

3.6 Evaluation of Articles

In this section, we discuss the study's evaluation of articles. We address each posed research question in detail and report the results accordingly. To address RQ-1 and RQ-2, we identified solutions proposed in the articles. These solutions are shown in table 3.4 along with their implementation tool support.

Table 3.4 Solutions proposed in articles

Ref.	Contribution	Contribution name	Implementation tool
[135]	Framework	STS-ml Extension	STS-Tool
[136]	Approach	Autofocus Extension	AUTOFOCUS Tool
[137]	Framework	UML based business process-driven framework	UML
[138]	Technique	Hazop	UML Use case Model
[139]	Technique	SRL	UML
[140]	Framework	Secure Tropos	ST-Tool
[141]	Process	SREP	CARE Tool
[142]	Approach	Trust Assumptions	Use of Trust Assumptions embedded in the solution
[143]	Approach	Secure Tropos Extension	Off-the-Shelf LPG-td Planner
[144]	Method	MOQARE	Misuse Tree
[145]	Process	Agent Oriented Process	Meta Agents
[146]	Approach	Scenario Driven	Conceptual Model
[147]	Framework	Three Layer Security Analysis Framework	Goal modeling
[148]	Process	Security Ontology	SQWRL
[149]	Method	Conceptual Framework	Reference Implementation
[150]	Framework	Parmenides	NFL
[151]	Method	UMLsec	UMLsec
[152]	Approach	Modular Approach	Secure UML
[153]	Framework	Extended previous framework	CONCHITA Tool
[154]	Approach	SURE	ASSURE
[155]	Method	i* framework Extension	si* Tool
[156]	Method	STPA	Rodin Toolset
[157]	Process	Security Development Lifecycle	Microsoft Tool
[158]	Method	SQUARE	CASE Tool
[159]	Framework	CORAS	XML Tool
[160]	Process	CLASP	CLASP Tool

To address RQ-3.1 and RQ-3.2, we identified security goals, threats and vulnerabilities mentioned by the researchers in their articles. Table 3.5 describes the security goals, threats and vulnerabilities in cyber-physical systems. The comprehensive list of security goals and threats has been provided in Chapter 5.

Table 3.5 Security goals, threats, and vulnerabilities in CPS

Security Goal	Layer	Threat	Vulnerability
Security goals aim is to protect the system from threats and vulnerabilities and reduce risk. Important security goals include Confidentiality, Integrity, Availability and Authentication	Physical layer	Physical attack, Natural disaster, DoS attack (Jamming), Sensor node compromising, Node-Replication	Platform configuration, Platform hardware, No physical protection for sensor/actuator
	Network layer	DoS attack, Eavesdropping, Compromised-key, Man-in-the-Middle attack, Wormhole attack	Network hardware, Network parameter, Network communication, connectivity
	Application layer	Malicious software, Unauthorized access, Data manipulation/ tampering, Social Engineering	Operating system and software patches are not up to date or not maintained

To address RQ-4, we identified the articles that were evaluated empirically as shown in figure 3.2. We found that 45% (140/313) articles were not empirically evaluated. Furthermore, we found that for empirical evaluation researchers used case study as the most used empirical method in articles 42% (132/313) followed by controlled experiment 9% (29/313), survey 2% (8/313) and other evaluation 1% (4/313) respectively.

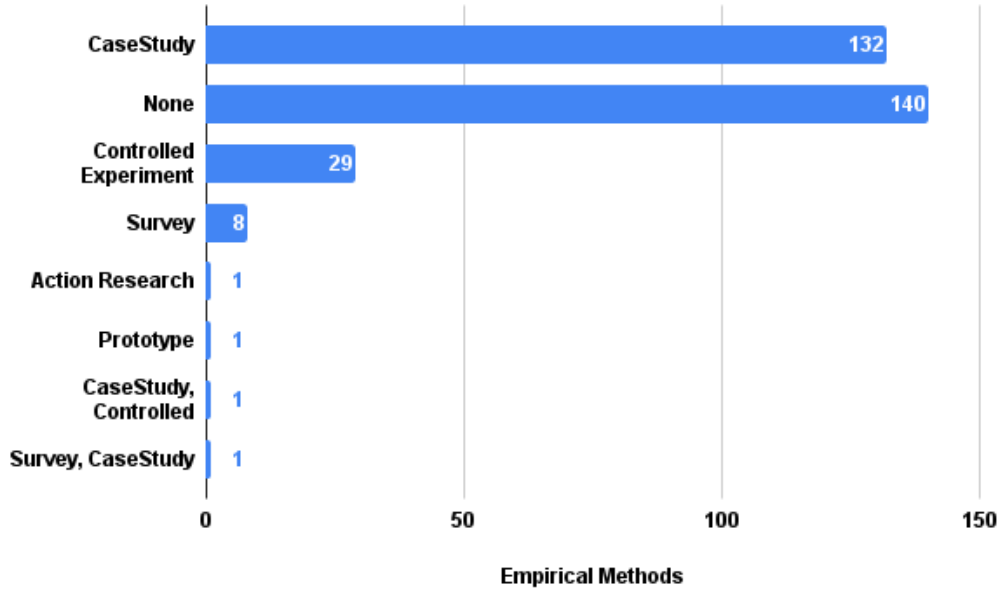

Figure 3.2 Empirical methods used for evaluation in selected studies

3.7 Chapter Summary

The study provided fruitful insights regarding pragmatic security requirements engineering solutions and their implementation tool supports proposed by the researchers. Moreover, the results reported in the study revealed that the trend of research interest is increasing in the domain of security requirements engineering especially for cyber-physical systems. The study would help its readers to better understand the techniques being employed to ensure security and the threats and vulnerabilities that might affect the system operations.

CHAPTER 4

Security Requirements Engineering Frameworks

In the field of Security Requirements Engineering (SRE) for software systems there exist a number of well-studied security requirements engineering frameworks, though none of them is considered the standard framework. Analysis from different organizational experts and practitioners indicates that the security requirements framework as a tool enhances the secure system development methodology that incorporates with any Software Development Life Cycle (SDLC) process [9]. This chapter discusses the purpose, scope and activities of a set of well-known security requirements engineering frameworks. Moreover, this chapter also focuses on a comparison of different security requirements engineering frameworks and illustrates the results.

Many organizations have already arrived at the conclusion that it is imperative to address security requirements earlier on in the lifecycle process. It has thus become an emerging focus area for researchers and an assortment of methods and tools are being developed. Different organizations, particularly giants like Microsoft have already incorporated SRE methods into their lifecycle process. There exists no consensus at this stage however, on a single best approach [161] [162].

4.1 SQUARE

SQUARE, the Security Quality Requirements Engineering framework offers an approach for analyzing and eliciting security requirements during the Requirements Engineering (RE) phase of system development [163]. It focuses on assessing threats and associated risk with reference to security objectives, for which different stakeholders ranging from user groups to service providers and security experts to requirements engineers are involved. SQUARE suggests and facilitates selection of existing elicitation, risk assessment and categorization techniques, though it does not

integrate security-related modelling techniques. The nature and identity of techniques to be employed in the security requirements phase is determined by the project team [164] [120].

The framework consists of nine activities and these include:

1. Agreeing on definitions: In this activity, the stakeholders and requirement analyst establish clear communication and agree on the specific definition of security in the context of the system.
2. Identifying security goals: The purpose of this activity is to set security goals properly and resolve the conflicts if any.
3. Developing artifacts to support security requirements definitions: in accordance with the definition of security, the following artifacts have to be developed: architecture diagram, use case diagram, misuse case diagram, attack trees and document template/form.
4. Performing risk assessment: This activity recognizes the vulnerabilities and threats associated with the system and applies risk assessment techniques.
5. Selecting elicitation technique: There are a number of elicitation techniques available and requirement analysts can choose any of them according to their suitability to the project. For example, to conduct an interview, develop a scenario or take the help of an issue-based information system.
6. Eliciting security requirements: After applying the elicitation technique, the security requirements are elicited.
7. Categorizing requirements: In this activity, the elicit requirements have to categorize using the following criteria: essential requirements, non-essential requirements, software level requirements and system level requirements.
8. Prioritizing requirements: After categorizing the requirements, this activity mainly focus on requirement prioritization according to the importance of security requirements.

9. Inspecting requirements: This activity ensures that all requirements are correct, feasible and implementable without any ambiguities, inconsistencies and mistake assumptions after the consultation of stakeholders. This result the final security requirements document.

All above activities have been performed using some exit procedure, which has to be met before entering into the next activity. SQUARE is well-known for supporting security goals such as confidentiality, integrity and availability (CIA). It forms a multilateral approach to the system and includes threats, risk assessments and quality assurance [120]. SQUARE properly addresses the conflicting requirements and validates elicitation requirements but it does not address asset identification and vulnerabilities to the system. It also does not explicitly mention the domain where the system operates [9] [120] [127]. It has been reported [123] [165] that SQUARE is used by certain organizations and SQUARE is validated in both academic and industrial contexts. SQUARE framework is one of the current SRE methods that are used to some extent in organizational contexts [9].

4.2 Microsoft SDL

Security Development Lifecycle (SDL) methodology was established by Microsoft to address security concerns that arise during software development. Microsoft SDL is a framework that is involved in all phases of the software development lifecycle. By adding security focused activities to each phase, security related deliverables are produced. Before moving from one phase to the next, a series of prerequisite activities must first be completed. Security is considered along with the functional requirements through the duration of the development lifecycle and security measures can be integrated in the design and architecture of the system. The operational efficacy of these measures is verified in the testing phase [166]. A security team is constantly involved during the development, which is a key characteristic of Microsoft SDL. As a member of the security team, the security advisor stays in contact with the

development team to offer support regarding the activities and processes that concern security aspects [157].

Microsoft SDL outlines seven phases that follow the development lifecycle. 1. Training: Knowledge and training are imperative for security matters. Where necessary, a core training and an additional training of the engineers is conducted. 2. Requirements: along with functional requirements the team identifies key security objectives. 3. Design: The security structure needs to be defined and the threats need to be modelled. 4. Implementation: During the implementation of the system the team applies coding guidelines, best practice, analysis tools and other techniques and standards to ensure that vulnerabilities are addressed comprehensively. 5. Verification: Implementation is rechecked and empirically verified to meet the security and privacy requirements for the system. 6: Release: A final security review is conducted to determine whether the system complies with all activities and security standards prescribed by Microsoft SDL. 7: Response: After the release of the software, Microsoft continues to identify and monitor security related incidents and responds accordingly [167].

SDL has very large methodology and is suitable for large-scale projects. Therefore, SDL covers various application areas that include business, personal and sensitive information, and web applications. SDL is not appropriate for small organizations because of its comprehensive methodology and it is very time consuming for organizations [168].

Figure 4.1 Microsoft SDL phases **[168]**

4.3 UMLsec

UMLsec is a modelling approach based on the Unified Modeling Language (UML), which is familiar to most developers, making it a convenient choice as a SRE platform. UMLsec extends the modelling aspects of UML by introducing security context features [169]. These features support the analysis and evaluation of models for security issues and vulnerabilities. UML models are extended amongst others with stereotypes, tags and constraints in order to formulate and support the definition of security requirements and security assumptions [170]. Additionally, UMLsec provides a notation for modelling threat scenarios and capabilities of attackers. The features introduced by UMLsec allow experts to apply security-focused analysis on UML models and regard security mechanisms separately from the core functionality [14]. Models created with UMLsec can be analyzed and validated by applying formal methods. Methods such as first order logic or formal semantics facilitate the detection of security vulnerabilities and enable verifying the models by automated tools. The use of UMLsec is not limited to particular development phases or system domains. Therefore, UMLsec is applicable to the requirements engineering phase as well as during the system design or verification phase [7].

UMLsec was developed by UML to establish security for critical systems. UMLsec is an extension of UML with the inclusion of security features. The methodology focuses on the development of design models with security features. The objective of UMLsec is to reduce time and cost in the development phase. UMLsec analyses and represents the security environment using misuse cases, class diagrams, activity diagrams, state-chart diagrams, sequence diagrams and deployment diagrams. UMLsec delivers three extensions to UML diagrams, i.e., stereotypes, tagged values and constraint [7]. It also includes the concept of an adversary who can harm the system. The main security requirements like secrecy, integrity, authenticity, secure information, etc. are defined using UMLsec notations. The purpose is to develop a secure environment for networking and the smooth functioning of security-critical systems. The methodology analyses the security at a fairly low-level and is suitable for an operational analysis. However, the methodology does not focus on elicitation, completeness or validation of security requirements [122] [120].

4.4 Secure Tropos

Secure Tropos extends the methodology of Tropos. Secure Tropos is a well-defined methodology to allow analysts to consider security issues at all stages of software development [125]. This gives developers the advantage of being able to apply the security at any stage of software development. Secure Tropos methodology consists of four stages [171] [121]:

- *Early requirements:* Deal with defining security requirements at an early stages of software development. The purpose here is to define and understand the problems of the existing organizational setting.
- *Late requirements:* Deal with analysing the system context in its operational environment.
- *Architectural design:* Deals with defining the system global architecture in terms of its subsystems.

- *Detail design phase:* Deals with identifying the security requirements at the detail level of system components.

To allow the analyst to clearly define security requirements, the idea of constraints is introduced, and the concepts of actors, goals, tasks, resources and soft goals are extended [172]. Secure dependency, security constraint and secure entities are introduced to help modelling security. However, the methodology does not support risk assessment and assets specifically [122]. On the other hand, Secure Tropos does not specify assets, thus security goals are related only to specifying the system goals. In ISO 27005, the asset identification is a main activity as the later phases use the valuable assets to evaluate and to protect them. Accordingly, Secure Tropos is clearly incompatible with ISO 27005 [165].

4.5 CLASP

Comprehensive Lightweight Application Security Process (CLASP) is a set of best practices that delivers a well-organized methodology to progress security in the initial stages of software development [173]. It identifies a set of processes that can apply in any software development. It brings an extensive set of security resources that make implementing activities realistic. CLASP implement the activities very realistically because of its extensive set of security resources. It comprises of 24 activities and five categorized views as shown in figure 4.2 that can be easily adopted into any software development process. These activities are described generally from a theoretical perspective and have a broader range of scope. The selection of the activities and execution order are left to the practitioner's discretion to make the processes more flexible and efficient [174].

CLASP is comprised of a set of role-based processes which assign to each person individually. These roles are important for software security and these roles help to organize the activities. Furthermore, roles are responsible for the outcome and maintain the quality of activities. CLASP contains comprehensive knowledge base

information about number of classes of vulnerabilities that identify the security concerns that arise in the development phase. CLASP is a lightweight process and suitable for small organizations having a lesser amount of security stresses. The main issue of CLASP is that the activities are defined in a very broad scope which is very difficult to adjust. CLASP methodology is far more effective as compared to ad hoc treatment of security requirements [160].

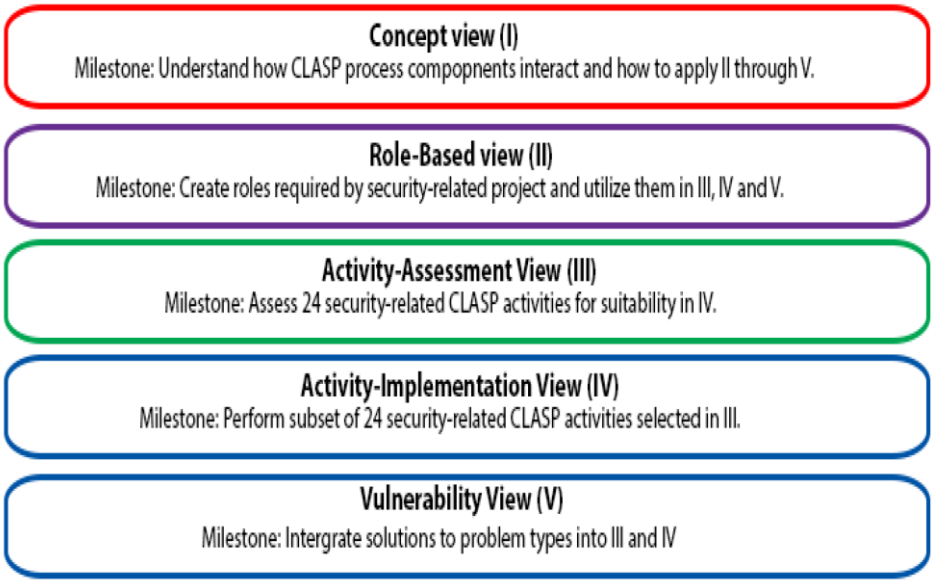

Figure 4.2 CLASP View **[160]**

CLASP methodology is based on a structured approach to develop secure software. CLASP is organized into seven best practices to enhance security, namely: (1) Institute awareness programs, (2) Perform application assessments, (3) Capture security requirements, (4) Implement secure development practices, (5) Build vulnerability remediation procedures, (6) Define and monitor metrics, and (7) Publish operational security guidelines. Moreover, CLASP has 24 practical activities that are closely connected to the best practices as defined above. Each activity is decomposed into further elementary steps, i.e., sub-activities. These activities can be incorporated into any type of software development methodology. This permits organizations to adapt CLASP in accordance to their needs [175].

4.6 Security Requirements Engineering Process

The Security Requirements Engineering Process (SREP) is a risk-driven and an asset-based methodology to define the security requirements. SREP is an iterative and incremental methodology based on Unified Process (UP) software development life cycle having different stages, as detailed below. SREP incorporates a set of security standards, particularly Common Criteria (CC), System Security Engineering Capability Maturity Model (SSE-CMM) and a number of ISO/EIC that focus on security engineering. SREP features the reuse of security elements stored in a Security Resource Repository (SRR). This is based on the idea that most security elements are applicable to multiple projects and that common elements, such as assets, threats, security measures or security requirements, could be reused [176]. SREP employs an iterative and incremental cyclical software development framework known as the Unified Process (UP) to evolve security requirements. With each cycle focusing on different aspects of the application's security, security requirements that cover various security risks can be specified.

SREP mainly consists of nine activities [141] [177]:

1. Agree on definition: The analyst team has to set the security definition, organizational security policies and the security vision of the system. This creates the security vision document, which describes the important information.
2. Identify critical assets: After the analysis of functional requirements, the important assets have to be identified.
3. Identify security objectives: In this activity, the security objective for each asset has to be defined, based on organizational policies. The list of security objectives shall grow and be refined in the ensuing iterations by creating dependencies between the security objectives.

4. Identify threats and develop artifacts: This step clearly defines the threats and contributes to developing the artifacts accordingly, i.e., use cases, misuse cases and attack trees.
5. Risk assessment: The probability of each threat and its respective potential impact of risk needs to be determined. The output should be captured in the risk assessment document.
6. Elicit security requirements: This activity elicits the security requirements. Each security objective is analysed for possible relevance and threats it poses. The output would come in security requirements specification document.
7. Categorize and prioritize requirements: This activity categorizes the security requirements into essential and non-essential requirements and prioritizes the requirements according to importance.
8. Requirement inspection: This activity uses the common criteria assurance requirements to validate the security requirements.
9. Repository improvement: The security resources repository should be added with some new elements. This is the additional activity of SREP methodology.

4.7 CORAS

CORAS is a model-based methodology to analyse security risk. CORAS uses a graphical or model-based approach and uses UML for modelling the security risk [178]. CORAS methodology offers a computerised tool to support documenting, maintaining and reporting analysis results through risk modelling, table-based documentation, consistency checking etc [179]. CORAS does not focus specifically on security requirements but is heavily oriented towards risk assessment. The developers must however, first understand the security requirements needed for risk assessment. Furthermore, the methodology also does not cover elicitation and validation of security requirements [122].

CORAS is divided into three different components:

1. It comprises of graphical syntax in CORAS diagrams and a textual syntax and semantics.
2. CORAS methodology describes step-by-step security analysis process with complete guideline of constructing the CORAS diagrams.
3. CORAS offers a tool to document, maintain and provide the results of reporting risk analysis.

CORAS methodology is divided into the following seven steps:

1. Introductory meeting: The first is called the introductory meeting. The main item of the agenda is discussed, and the analyst gathers data from the customer(s). The customer presents the overall goals of the analysis and objective they would like to have.
2. High level analysis: The analyst explains their understanding from the first meeting and resolves any ambiguities in the information they have received from the customers. They analyse threats, vulnerabilities, threat scenarios and any unwanted scenarios.
3. Approval: The target description is enhanced and the assumptions and preconditions are analysed. Both parties make final decisions on the asset and rank it according to importance.
4. Risk identification: The step starts with a brainstorming session. All stakeholders gather to contribute their experience and identify possible threats, vulnerabilities and unwanted risk incidents. The identified risks are documented and risk evaluation is created.
5. Risk estimation: This step is also organized as a workshop. This activity focuses further on estimating the consequences and probability values for each of the identified adverse events.

6. Risk evaluation: A possible risk matric is presented to the customer. The customer prompts few adjustments and modifications.
7. Risk treatment: This step is devoted to treatment identification and address the cost/benefits issues for treatments.

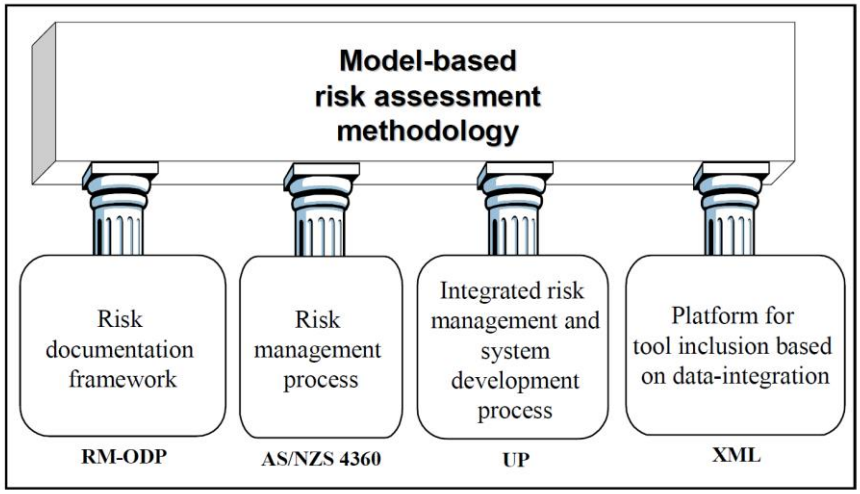

Figure 4.3 CORAS Framework [159]

4.8 Comparison of Security Requirements Engineering Frameworks

In this chapter, we review the most commonly used security requirements engineering frameworks. The comparison of SRE frameworks starts with the investigation of literature review. The best papers from the related to the SRE domain have been selected following a thorough search from the literature. Consideration has been given to the basic and important security elements/activities that are common across all the frameworks. Our comparison of frameworks uses the criteria defined in the literature [127] [165] [120] [9]. We also added the criteria according to the definition of CPS, a critical element of which is the physical environment, differentiating it from classical software. This set of criteria helps in comparing different SRE frameworks with each other. We determined 13 criteria:

- Stages of SDLC: To know the stages of software development life cycle.
- Definition: The analyst team has to set the definitions of security, threat and other important security related terms.
- Domain knowledge: Knowledge about which environment the system would operate in.
- Stakeholder views: Taking the opinion of all stakeholders.
- Security goals: It fulfils the concepts of: (i.e. confidentiality, integrity, availability etc).
- Assets: Takes into consideration the value of stakeholder's assets.
- Threat: Takes into consideration anything that may harm the system.
- Misuse case: Utilizes the misuse case technique to analyze threats.
- Vulnerability: Analyses weaknesses in a system.
- Risk: Analyzing the various possible risks to the assets due to different threats.
- Categorize & prioritize: To categorize and prioritize the security requirements according their importance.
- Validate: Includes a step of validation of security requirements.
- Physical environment: Takes into consideration the physical environment of a CPS including sensors, actuators and gateways.

Table 4.1 Comparison of SRE Frameworks

Frameworks / Criteria	SQUARE [123]	MS SDL [168]	UMLsec [7]	Secure Tropos [172]	CLASP [160]	SREP [141]	CORAS [159]
Stage of SDLC	RE	Entire	Design	Entire	Entire	RE	RE
Definition		x	x			x	
Domain knowledge					x		
Stakeholder views	x	x	x			x	
Security goals	x		x	x		x	
Asset		x	x		x	x	x
Threat	x	x	x		x	x	x
Misuse case	x	x	x			x	
Vulnerability				x	x	x	x
Risk	x	x			x	x	x
Categorize & prioritize	x	x	x	x	x	x	
Validate	x				x	x	
Physical environment							

From the table 4.1, it has been seen that few security requirements frameworks are applied to the entire development life cycle for software like Microsoft SDL, secure Tropos and CLASP, each proposing a number of different activities across the complete development life cycle in order to increase the security. Therefore, these frameworks applied throughout on the development life cycle and cannot be applicable on a single phase. Few frameworks are applied only on requirements engineering phase, for instance SQUARE and SREP. These two methods are applicable only on requirements engineering phase to enhance the security while UMLsec is only applicable on design phase to cover the security aspects.

We found some limitations of each framework. SQUARE framework is mainly developed for security requirements engineering phase. The framework does not focus on domain knowledge and not explicitly/implicitly mention asset and vulnerability. On the other hand, SREP is quite similar to SQUARE with some additional component like repository, where similar project history can be added on to this new repository. Microsoft SDL covers all the development life cycle but does not focus on security goals and domain knowledge. Similarly, CLASP and Secure Tropos also covers entire development life cycle but does not focus on stakeholder views and misuse cases. UMLsec framework is specifically applicable only on design phase but does not include domain knowledge and risk assessment. The main emphasis of CORAS is risk related factors and does not focus on misuse case, nor does it categorize and prioritize the security requirements.

It is evident that every framework holds certain advantages and disadvantages and is well-suited to a specific purpose. Furthermore, there is no standard security requirements framework that fulfils all the needs of every organization. Therefore, this variety of options poses a difficult choice to security analysts to select an appropriate security requirements engineering methodology according to their needs and expectations. Furthermore, these frameworks were specifically designed only for software systems and not for CPS, as none of them focus on the physical environment which is an important component of CPS. Given the fact that CPS are tightly coupled to the physical environment, and interact with it directly by means of sensors, actuators and gateways, it is imperative for any CPS security framework to address the security of the physical layer. In this case, merely relying on standalone physical layer protections provided by manufacturers do not provide holistic security in light of their functions in the CPS. A detailed discussion regarding the nuances of CPS as opposed to classical systems and the need for particular attention to be given to physical layer security within a CPS can be found in Chapter 2. Therefore, it is very important to explore some new security requirements frameworks, especially for CPS.

This comparison helps us to determine the strengths and weaknesses of each framework. Our findings from this comparison survey indicate that none of the frameworks perform all the required activities for secure cyber-physical systems. Furthermore, this comparison helps us to identify the shortcomings in security requirements engineering frameworks which have been rectified in our proposed security requirements engineering framework for cyber-physical systems.

4.9 Chapter Summary

In this chapter, well known security requirements engineering frameworks are examined and compared with each other. The security requirements engineering frameworks are developed to enhance the security of software such as SQUARE, Microsoft SDL, UMLsec, Secure Tropos, SREP and CORAS. These frameworks provide guidance such as security definition, domain knowledge, security goals, threat analysis, validation etc. But there is not a generally accepted framework for the development of secure CPS. Here we have taken a look at the strengths and weaknesses of each framework. Our findings indicate that none of the frameworks fulfil all the desired functionality expected of secure CPS. Using the shortcomings seen from this comparison, we are able to identify the elements that need to be rectified in SRE frameworks, which has been followed in our proposed SRE framework for CPS.

CHAPTER 5

Proposed Security Requirements Engineering Framework for Cyber-Physical Systems

This chapter proposes a security requirements engineering framework for cyber-physical systems. The purpose of this proposed CPS framework is to introduce the concept of early security in the requirements engineering phase. Our framework is a representation of underlying processes to elicit the security requirements for cyber-physical systems. The proposed CPS framework comprises of three main components: analyzing the CPS environment, conducting a defined set of security requirements activities and employing misuse case technique. The activities and misuse case technique help to elicit the security requirements for cyber-physical systems. For each activity, we present the security concepts and explain their significance and contribution to the framework. Then we explain how to apply these activities in the project. Moreover, this cyber-physical systems framework can serve as a basis and detailed guide for practitioners and researchers working towards determining security requirements by outlining the set of essential activities required in the process. In order to practically implement this framework, we have also developed a tool that can be used to systematically perform the activities mandated by the framework. At the end of this chapter, we present a step-by-step illustration of how this tool is to be implemented.

5.1 Overview

In an age when all things have become increasingly digitized and automated, cyber-security has become more important than ever. With this in mind, it is no longer sufficient to just make use of a general purpose security system for complex cyber-physical systems, and so, the field of Security Requirements Engineering (SRE) comes into play [180] [181]. Security requirements engineering can be described as the process of eliciting, analysing, specifying, and validating the security requirements of

a system [9]. Unfortunately, it tends to be overlooked during the requirements engineering process, or even when taken into consideration, the security requirements and the main system requirements are considered as separate entities of unequal importance, and thus not integrated together into a coherent whole. Conventionally, a predefined set of standard security requirements have been generalized to be applicable to all systems, and with this set of security requirements being seen as a sufficient base from which to implement a blanket security system, an acute analysis and specification process is not conducted. This results in weak security requirements specifications, which adversely affect the design and implementation of a system. Security requirements analysis carried out in a standardized and systematic manner can add significant value to a system, particularly when it is integrated in the early stages of system development [21].

Security requirements have gained increasing importance in the development process of cyber-physical-systems. They aim essentially at developing the software to run continuously and accurately in every conceivable circumstance. These circumstances can be different in nature, starting with natural disasters or widespread power failures and leading to vandalism, terrorism or malicious attacks. Compared to classic systems which only include software, cyber-physical systems also offer a new, more corporeal dimension so to speak, the bridge to the physical world. Since, these physical gateways, the software itself and the communication between the two each offer possible avenues of attack securing them all is an essential requirement [21]. This has been discussed at length in Chapter 2.

5.2 Security Requirements Engineering for Cyber-Physical Systems

Security requirements engineering for cyber-physical systems is a field of security requirements engineering that focuses on security analysis and specification of security requirements for CPS, as mentioned previously. Cyber-physical systems have distinct characteristics that differ from traditional systems and hence require diligent consideration when building security into cyber-physical systems. Since software gets continuously more complex, vulnerabilities are rising. With cyber-physical systems posing additional vulnerabilities as a consequence of exposed physical gateways, it is not feasible to apply security as a set of auxiliary security measures to an already deployed system due to the complex relationships between the layers. Rather, it must be implemented in the beginning of the system development process. Security for CPS, just like other software systems, must be treated as a perpetual, life-long procedure in the developing process, starting with the conception and ending only when the system is no longer functional. The majority of the effort spent on security requirements should be invested in the beginning, i.e. Requirements Engineering (RE). Nevertheless, security issues may arise at any point in time, and the developer should always have a plan of action in place to effectively deal with them.

Security requirements engineering consequently aims at systematically building secure software by generating a complete set of detailed security requirements. These requirements could take a variety of forms, for instance "the data should not be diverted or transferred through the communication network", or "the sensor shall not divert the data into another data acquisition board/server", etc. Seeing that due consideration is not given to system security requirements, a large gap is left in the so-called Software Development Life Cycle (SDLC). This gap needs to be filled during the RE process. Since incomplete security requirements can lead to costly and risky consequences, particularly so for CPS, it is important to be aware of as many conceivable vulnerabilities and threats as is possible. Therefore, not only obvious

software and hardware security requirements must be created, but also requirements concerning the analyst, the users and many more which may not be plausible at the first look [21] [182]. Ideally, security requirements should be determined prior to the development of a system so that they may be used as a metric against which the end product may be gauged. Furthermore, the listed security requirements can be easily verified against each of the security features of system, creating a checklist of sorts to determine the completeness of the security elements. Therefore, our aim is to fill this gap and propose a SRE framework that determines the system security requirements in the RE phase.

5.3 Proposed Security Requirements Engineering Framework for CPS

The proposed SRE Framework for CPS aims to serve as a complete guide through a number of activities designed to analyze and identify threats as well as to determine security requirements of CPS by taking different aspects of CPS into account. The purpose of this security requirements engineering framework is to identify security requirements satisfactorily, prior to the implementation of the system. As previously mentioned, there is currently no comprehensive security requirements engineering framework available for CPS, since the nature of CPS is quite different to classical software systems because of the CPS characteristics of heterogeneity and adaptability that result from the addition of the physical layer. Therefore, we propose a security requirements engineering framework (SRE Framework) that consists of a set of essential activities required to elicit the security requirements for CPS. This quest leads to RE methodologies so that security concerns can be addressed during the early stages of software development. The proposed framework is a systematic approach to incorporate assets, security goals, threats, and risk assessment that are critical to the CPS. We propose a set of eight main activities, and utilize the '*misuse case*' technique proposed by Sindre and Opdahl [183] as shown in figure 5.1. A misuse case is operated like a use case while being essentially its converse, i.e., it pertains to a function that

does not permit the system to operate in a normal manner or state [183]. The process we have proposed is an iterative one. After analysis of the CPS environment, we conduct the eight activities detailed in our framework. These activities may reveal certain new information or place certain restrictions or requirements on the system, which would also have an effect on the nature and specifics of the CPS environment. This will prompt be revisiting the activities and take in the input from a possibly modified CPS environment.

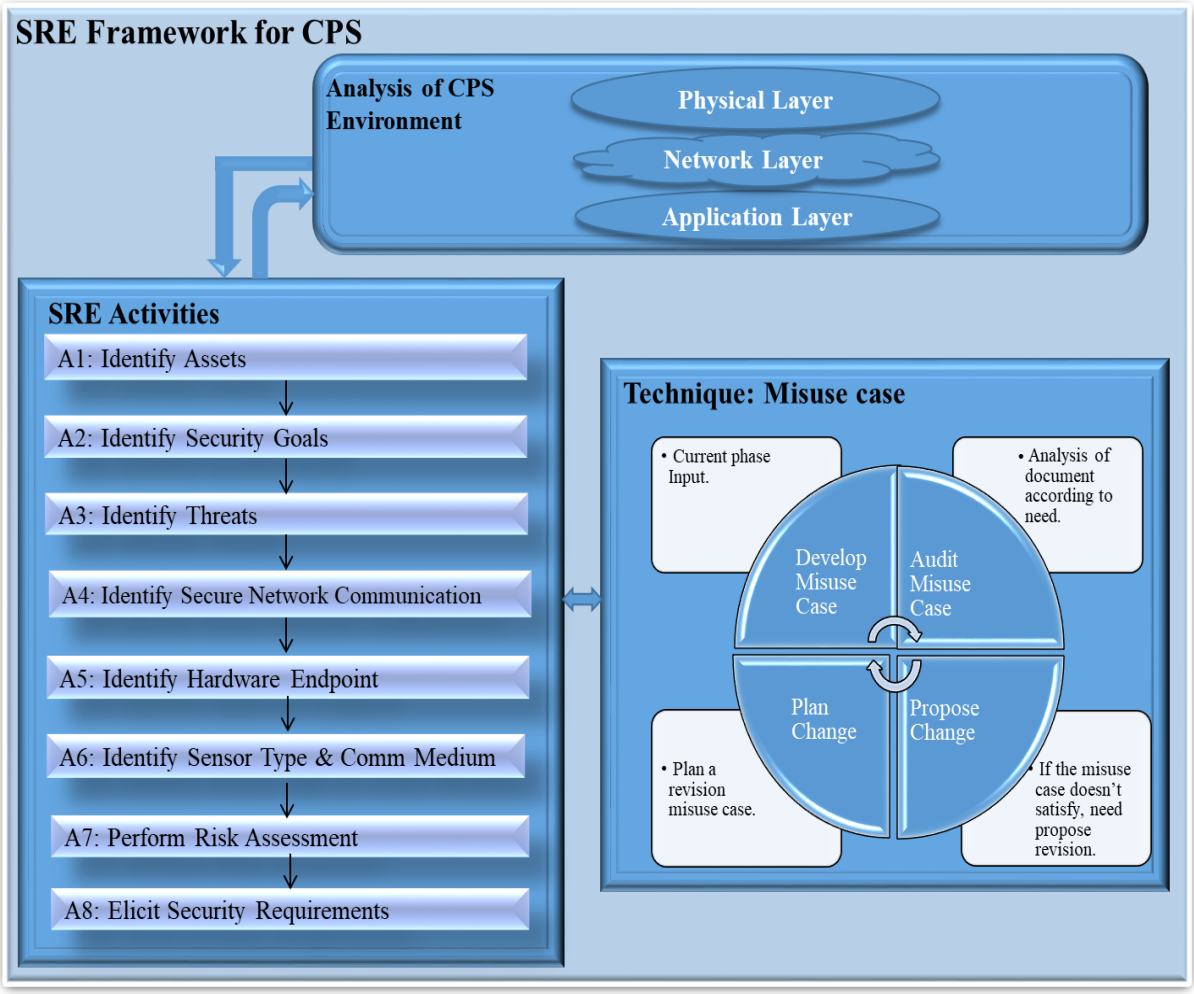

Figure 5.1 SRE Framework for CPS

The CPS framework consists of following three main components: Analysis of CPS Environment, SRE Activities and the Misuse Case Technique.

5.3.1 Analysis of CPS Environment

The purpose of this component is to analyze the system environment and to identify its cyber-physical systems properties. Here, the relevant system environment of the CPS is defined. The CPS environment is an important component that influences the understanding of the system and hence the definition and interpretation of requirements [184]. Accordingly, the CPS environment contains information that plays an essential role in the definition of security requirements [185]. In order to identify the CPS environment, aspects in the system's operational and physical context are analyzed. Relevant aspects range from users, external systems that interact with the system to existing processes, components and subsystems that shall be embedded [186]. Furthermore, context analysis also includes considerations about phenomena that occur in the physical environment as well as restrictions due to resources, organizational rules, legislative regulations, etc. Cyber-physical systems consist of multiple heterogeneous devices and systems. This results in the fact that particulars of cyber-physical systems differ from system to system [187]. However, each CPS shares some common architectural characteristics such as the presence of three layers, vast communication between the sensors, network and actuators, etc.

Analysis of the CPS environment begins with use cases for the system. Use cases give the system functional requirements, which help in designing the initial high-level CPS architecture. The architecture gives us an idea about the components of the CPS, the external environments that it is in contact with and the relationships between the two. After analyzing the system environment, we perform the framework activities, and we are able to update our system architecture to reflect the new decisions we have made, as the proposed process is an iterative one. This in turn, allows us deeper insight into the system environment and its specifics, as well as giving us more exact inputs for the framework activities. We continue to proceed and refine until we arrive at a reasonably sufficient understanding of the CPS environment and as a result, a fully defined set of security requirements.

The environment consists of different aspects that relate to the system under consideration, for which a sufficient understanding is expected. The analyst should begin by researching organizational processes, physical and software components, external systems that interact with the CPS and the nature and basis of their interaction, legal and regulatory restrictions, codes and conduct, standards, safety considerations, system users and stakeholders. It is important for the analyst to understand the system usage. The clear and necessary understanding is an important aspect of system usage.

The analyst must look through the operational and technical environment for policies and strategies outlining restrictions or guidelines for using any technology or operational environment. For example, in the case study performed with Soccerwatch, upon detailed prompting, it was revealed that the company had certain collaboration agreements with the Vodafone data provider, and so there were already restrictions in place that mandated the use of 4G communications for the CPS. Analyzing this policy requirement was a critical factor in determining security requirement number 26.

At the implementation level, many times assumptions need to be made about the environment. These assumptions relate to factors in the environment that cannot be foreseen to a high level of certainty, such as average amount of traffic expected on the system. Since availability is a key security goal, the CPS security requirements analyst should make an assumption beforehand about the expected number of visitors at any one time, and ensure smooth running of the system to a reasonable factor of safety. This assumption about the environment led to security requirements 41 and 42 from the Soccerwatch case study.

Assumptions should also be made about the physical environment and its uncertain future state. For instance, when choosing electronic sensors that are sensitive to ambient temperature, certain assumptions about the locations where the CPS can be

installed, the season of the year they will most likely be used in, and the average forecasted temperature and humidity at such place and time must be assumed to arrive at optimal decisions about hardware endpoint selection. This is particularly true where mild changes in temperature may lead to integrity of components being compromised. This consideration led us to the security requirement number 12 from the Soccerwatch case study.

5.3.2 Security Requirements Engineering Activities

In Chapter 4, we presented the well-known and commonly used security requirements engineering frameworks established in the literature. After analyzing these frameworks and adding relevant activities that addressed security in the physical layer, we were able to list sixteen activities to be performed as part of the SRE process for CPS. These activities had been selected based on their importance vis a vis security requirements engineering [188] [120] [121] [9] [189] and in line with the definition of cyber-physical systems [1] [30] [190]. Upon further review, we were able to shorten this list to a set of eight activities that were entirely essential to the security requirements elicitation process, and which were free of redundancies. To aid in the process of identifying necessary activities that together could successfully and completely elicit security requirements of a CPS, we established a set of matrices. This was done by taking common security elements mentioned in the literature and setting them as criteria to judge the effectiveness of existing frameworks and their constituent activities. This set of matrices was applied iteratively to our proposed framework (using a case study model of our developed functional prototype of a smart car parking system) by which the critical activities were shortlisted.

During this process, we applied all basic and fundamental security elements and analyzed thoroughly all three layers of CPS for attack points, vulnerabilities and threats. This was instrumental in recognizing the essential activities to determine the security requirements of a CPS, without the presence of any redundant activities.

Therefore, it is recommended to apply all of the recognized activities to determine the security requirements for cyber-physical systems. The framework has the eight activities A1 to A8. Below we describe each of these activities and explain their importance and contribution to the framework.

5.3.2.1. How to Apply SRE Framework

Cyber-physical systems SRE Framework may be most effective when applied to a system under development and applied once the analysis of functional requirements has been accomplished. The proposed framework can be adopted into the stages of an organization's existing system development life cycle, especially during the requirements engineering phase. The SRE Framework process involves the interaction of a team of requirements engineers, the security analyst and the other stakeholders. Table 1 illustrates the way in which the proposed framework is applied to the CPS, which explains the input, technique, output and name of the activity. The framework is in the form of a checklist that needs to be followed. The output from each activity represents its completion. The framework proposes an agile methodology to select the required activity. Each activity contributes to a collective whole, and its results can be used for both the preceding and succeeding activities. Where applicable, it is also recommended to follow the security standards (e.g. ISO/IEC 27000, ISO/IEC 27001) [191] [192]. The analyst needs to review each activity and may propose changes if and where required. Having completed the sequence of prescribed activities, the analyst will be able to obtain a set of the desired security requirements for the CPS.

Table 5.1 Framework Workflow Process

Activity	Input	Technique	Output
A1: Identify assets	Architecture, generic checklist of assets	Use case, facilitated meeting sessions	List of CPS assets
A2: Identify security goals	Output list of CPS assets, generic checklist of security goals	Facilitated meeting sessions (detail analysis, interview)	List of security goals
A3: Identify threats	Generic checklist of threats, Output list of assets & security goals	Misuse case, questionnaire	List of CPS threats
A4: Identify secure network communication	Generic list of network communication, protocol	Facilitated meeting sessions (analysis and comparison)	List of secure network communication
A5: Identify hardware endpoint	Checklist of hardware endpoint	Facilitated meeting sessions (analysis, group discussion)	List of hardware endpoint
A6: Identify sensor types and communication medium	Generic types of sensor, checklist of sensor communication medium	Facilitated meeting sessions (analysis and comparison)	List of sensor types & sensor communication medium
A7: Perform Risk assessment	Output list of assets and threats	Risk Matrix and Misuse case	List of risk
A8: Elicit security requirements	Output of all preceding activities A1 to A7	Facilitated meeting sessions (analysis, group discussion), misuse case	List of security requirements

A1: Identify Assets

An asset could be anything that has value for the organization i.e., people, money, software, hardware, sensors, etc. Therefore, the purpose of this activity is to determine all the assets of the CPS components. This activity also involves an evaluation of environmental and organizational assets. These assets usually involve human resources, data resources, network resources, and sensors and physical components. In order to identify assets, the CPS is analyzed for valuable objects that are potential targets for an attacker. A characteristic of an asset is, that in case that it is obtained or exploited by an attacker the system might suffer a loss. The goal of this activity is to identify application assets, network assets and physical assets, e.g. sensors, servers, gateways, etc. Results from preceding activities, i.e. functional requirements and stakeholder opinions, and additional resources, such as documentations of common assets in CPS, also facilitate identifying the systems assets.

Some assets are more important than others, it makes sense to classify them with respect to the value they possess. For instance, a special operating system or expensive camera probably have more value to the system than the web server, which is easily replaceable. Software components that the system depends on like Kerberos, a third-party authentication program, for instance, or operating system environments like Microsoft Windows can be considered as software assets. Furthermore, assets may include a range of hardware devices (i.e. sensor, camera, mechanical supports, etc). Finally, stakeholders - human-beings or organizations with influence on the system - themselves are significant assets to the CPS. These can be of two types, those who are involved in the project, like developers and the management, and those who are affected by the project and will use its artifacts. Data that holds importance to the operation of the system or to its stakeholders is also counted as an important asset.

Assets are elements of the target of analysis that hold some value to the client and which the client wishes to protect. This could be in the form of physical objects, key personnel, services, software and hardware. Relatively intangible things such as information, expertise, trust, market share and public image may also be included in the sphere of assets. Input assets to this activity, in terms of a CPS are shown in table 5.2. The analyst team must select the asset according to their needs and the CPS environment. The identification of asset is supported with the use of pre-defined checklists and extra assets can be added according to user needs by using any technique discuss below. The architecture of CPS environment would also help to identify the assets. The analyst team can select the desired assets from the checklist.

Input: A general checklist of CPS assets are listed in table 5.2 [1] [193].

Table 5.2 Checklist of CPS Assets

Application program	Software program designed to fulfil a specific purpose.
Customer data	Customer personal data.
Organization data	Organization personal data.
Software services	Customers access software over the Internet.
Data files	Transactional data having all information.
Server	Which hosts one or more websites/databases.
Gateway/Sink node	Serves as the connection point between the physical environment and controller.
Sensor	To perceive events in the environment and send the information to gateways.
Actuator	A device to operate the components.
Controller	The main processor to control and manage devices.
Smart meter	A measurement device.
Point-of-Sale terminal	A computerized device.
Admin/Manager	A person who is responsible to manage the IT system.
Stakeholder	Anyone who has interest in the project.
Antenna	It is used for electromagnetic/radio waves for communication.
Microphone	A device to record the voice or sound.
Monitor	Hardware to show the graphical user interface.
Computer	Programmable electronic device.
Camera	Camera for recording critical sections.
Radiator	A device to use for the emission of light, heat, or sound.
LED light	A light-emitting diode that glows lights when receives current.
LED display	A screen display technology that uses a board of LED.
User devices	A user devices can be computer, laptop, smart phone, tablet etc.
Transceiver station	To facilitates wireless communication between a device and network.
Fences	Fences to prevent intruders from getting access to physical objects.
Transformer	A machine to manage the voltage.
Security guards	Security guards to prevent intruders from getting access to physical objects.
Parking lot	A parking area used for vehicles.
Barrier	A control access by being raised or lowered.

Technique: When selecting a desired asset, it is useful to use certain techniques that will help the analyst team to avoid expensive mistakes and select required assets only, thereby saving precious time and effort. Following techniques are useful to select the required assets for CPS:

Use case: By creating a use case, the analyst team can analyze the assets and select according to its importance. By using this technique, the analyst and stakeholders can easily select assets.

Facilitated meeting sessions: The detail analysis can possibly be done by creating a group discussion session. The analyst team, security advisor and all stakeholders participate in this session and discuss all possible assets and then finalize the assets.

Output: After identification, the assets will be assigned a rating commensurate to their value or importance to the client – either with regard to the system operation, the business, system security, reputation, etc - so that the most important assets can be selected. This will help facilitate prioritizing the risks later on.

A2: Identify Security Goals

Business goals and quality attributes (performance, robustness, etc) are combined to develop the required security decisions. This is achieved to identify security goals. Security goals are a minimal set of objectives that when achieved, would fulfill the purposes of security in any given system. Security goals mainly refer to confidentiality, integrity, availability, and authentication. Security goals aim to protect the system from threats and vulnerabilities and reduce risk factors. We aim to extend our understanding of security goals for CPS. For instance, in the case of sensor-data oriented systems with multiple sensor nodes generating data, security assurance is crucial to confirm if the data generated is coming from a trustworthy source, i.e., authentication becomes a critical security requirement.

Security goals determine the level of security that shall be incorporated in the system. On one hand security goals are defined based on the identified assets of the system, and on the other hand express the stakeholder's conception of a secure system. This activity aims to provide a documented list of security goals, the purpose of which is that malicious actions due to vulnerabilities and threats shall be prevented.

Security goals get identified through the requirements elicited by the stakeholders and the developers. These requirements generally belong to some contexts and some assets. In this relation context means the physical environment in which the system runs, the people using it or the operating world the software runs in. In special cases, for instance, non-functional requirements (quality requirements), it is not so simple to associate a material context or asset. Since these requirements strongly differ depending on the current project, security goals are very special for each system. Nonetheless, the four main security goals confidentially, availability, integrity and authentication can be found in every security policy. Prior to identifying security goals, an analysis of assets was conducted. Therefore, security goals are identified based on the output of assets for the CPS project.

Input: The team of analysts selects the desired security goals from the checklist as shown in table 5.3 [194] [53].

Table 5.3 Checklist of Security Goals

Confidentiality	Limits access to information and similar to privacy.
Integrity	The information is accurate and trustworthy to the users.
Availability	Availability ensures that the information is available all the time to the authorized user when required.
Authorization	Only authorized sensors provide the information to gateways.
Authentication	Nodes (sensors) should be identified and authenticated before adding them to the network.
Nonrepudiation	Sensor node should not refuse to send the information.
Freshness	This ensure that the information is recent and actual.

Technique: Following techniques help the analyst team to finalize the security goal:

Facilitated meeting session: The detail analysis can be conducted by creating a meeting session. The analyst team, security advisors and all stakeholders participate in this session and discuss all possible security goals and then finalize the security goals.

Interview: Interview is a very valuable technique for gathering important information. Problems can be further investigated, triggering user views and recognising unexpected problems. Therefore, interview can also support to finalize the security goals for CPS.

Output: After employing a combination of the methods mentioned above, the analyst team can easily identify the final list of security goals and add it to the output.

A3: Identify Threats

The purpose of this activity is to identify the threats for cyber-physical systems. We have classified threats into 3 categories; application layer, network layer, and physical layer. Threats are enacted through attackers. Attacker refers to entities with malicious interests, such as harming the system or its users. This activity also analyzes potential attackers, their capabilities and motivations and to describe the risk they might pose to the CPS. In order to recognize attackers and their capabilities, stakeholders with potential malicious interests and external attackers, i.e. hackers and cybercriminals, are analyzed. Results from preceding activities, i.e. stakeholder opinions, assets, security goals, additional resources, such as documentations of common attackers facilitate identifying and describing potential attackers of the system. The result of this activity is a detailed description of the type, motivation, capabilities and malicious actions of each attacker the system is exposed to.

Threats of CPS are identified by analyzing the system for loopholes that might facilitate attacks as well as modelling security-critical aspects of the system. In terms of the latter, the security team makes use of security modeling approaches such as misuse cases. Results from preceding activities, i.e. assets and security goals serve as an input for threat identification. Additionally, the security team can refer to other information sources, for instance common attack patterns and descriptions for CPS.

Usually, the origin of threat is a malicious attacker who tries to fulfill some aims. These goals can be deduced by one or more motives, which can be of various natures. The

most obvious intention would be economically motivated. The attackers simply seek to gain financial benefit by breaching, taking control of or disabling the system. On the other hand, politically motivated attacks seem to have become more and more commonplace, not only in the case where the attacker is a group of individuals with an obvious disregard for the law, for instance, extremist groups trying to spread propaganda, but also whole states have proven to hold these kinds of motives [195]. The Stuxnet case is a perfectly relevant example of this sort of motive, where the USA government initiated an attack on Iran's nuclear enrichment program back in 2010 [196].

A threat may exploit a vulnerability in the system. Vulnerabilities present flaws in the system an attacker might attempt to exploit. The system is analyzed for these flaws that lead to vulnerabilities. In addition, other resources can be included in the identification process, for instance documentations of common CPS-vulnerabilities, stakeholder opinions and attacker descriptions. A threat may harm the system, whether it is in the form of physical damage or through malware. Furthermore, assets, being a store of value, are susceptible to a host of negative consequences with a large impact if a threat is realized. Therefore, when analyzing the importance of threats, extra attention needs to be given to the assets, proportional to their respective significance as an asset. In this regard, a threat could be anything that may harm the system. It could be a human threat or could be from a natural source or could be an unexpected system behavior (e.g. the Boeing 737 MAX problems). The human threat could be an intentional threat or an unintentional threat as shown in figure 5.2. An intentional threat is where the incident is caused deliberately, whereas an unintentional one could be in the form of human error.

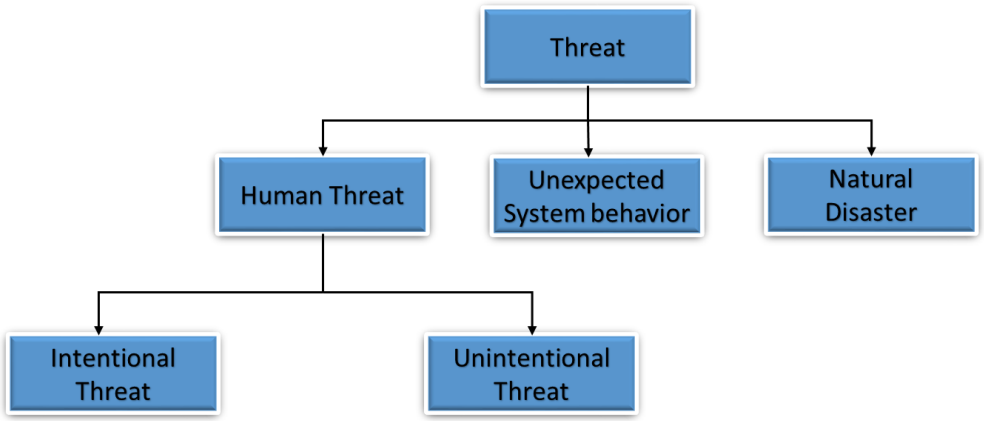

Figure 5.2 Threat category

In order to identify attacks that the system is prone to, assets, vulnerabilities and attacker descriptions combined were associated to threat descriptions. A threat description includes the type of attacker, his motivation for launching the attack, the malicious actions the attacker may perform on certain system components and the potential consequence if the attack is successful. The consequences in a threat description further show which CPS layer of threats may be enabled by the attacker's success. The identified threats illustrate that an initial successful attack can result in a state of one or multiple conditions that facilitate follow-up attacks. That means in order to achieve a major goal that an attacker is not able to achieve directly because, as in the case of unauthorized access, it is well protected by the system, the attacker launches a sequence of attacks that breaches the system's security mechanisms. The analyst team and stakeholders select the desired threat from the following checklist of table 5.4, 5.5 and 5.6. The results of this activity are major threat descriptions that include follow-up threats.

Input: A checklist of threats, categorized into three layers (i.e. application layer, network layer and physical layer) [19] [97] [197] [198] [111]:

Table 5.4 Checklist of Application Layer Threats

Unauthorized access	An attacker access the data in an unauthorized way.
Malicious software	Malicious piece of code which spreads from host to host using an infected file.
Spyware	Collects information about Internet search requests.
Ransomware	Encrypts data and asks for money to decrypt.
Phishing	Attacker masquerades as a legitimate person/entity (E.g. fake Facebook website to extract login details).
Password guessing	Attacker guesses the passwords of users.
Disclosure of sensitive information	Sensitive information can only be accessible to the authorized persons.
Repudiation	Deny of information for various reasons.
Identity theft	An attacker assumes a false identity, attacker takes advantage of data about another person, to act on attacker behalf.
Social Engineering	Tricking employees to win information/access to a system.
Key logger	Malicious software monitoring each keystroke typed.
Remote access	Fully access to the system in remote (likely combined with backdoor attacks).
Data manipulation / tampering	Data can be manipulated in several ways, e. g. by wrong data, any modification of database.
Dropper	Small file dropped into the system downloading more malicious software.
Drive-by-downloads	Automatic downloads when visiting a malicious website.
Elevation of privilege	A malicious user acquires a higher level of privilege to compromise or destroy a system.
Replay attack	Multiple, same looking login screens. The first-time data entered, it will be forwarded to the hacker.
Blue jacking	Sending unauthorized messages to Bluetooth enabled devices.

Table 5.5 Checklist of Network Layer Threats

Man-in-the-Middle	An attacker interconnects secretly between two parties and can listen to, replay or modify packets.
Eavesdropping	An attacker listens to messages.
Packet replay	An attacker replays or delays a transmitted packet.
Packet modification	An attacker modifies a transmitted packet.
IP spoofing	An attacker creates a new packet with a false source IP address.
MAC spoofing	An attacker changes the Media Access Control address.
Denial of Service (DoS)	An attacker floods the bandwidth or resources of a system (The system cannot response to legitimate users).
SQL injection	Illegal SQL query's over input fields for execution (e.g. database)
Routing attack	An attacker spoofs, alters or replays routing information.
Distributed DoS	Multiple attacking systems performing a DoS attack.
Wormhole Attack	An attacker uses a malicious node to redirect data received to another location in the network.
Slowloris	Holding many connections to a target server until it crashes.
HTTP flood	Flooding a target server with GET and POST requests.
Teardrop	Sends fragmented packets to a target machine which cannot reassemble them → Target system crashes.
Sniffing	Software, which monitors all network traffic.
Bluesnarfing	Exploiting information from a wireless device through a Bluetooth connection.
Compromised-key attack	An attacker figures out the secret key used to encrypt/decrypt data.
Remote access	Fully access to the system in remote (Likely combined with backdoor attacks).

Table 5.6 Checklist of Physical Layer Threats

Node-Replication	An attacker can add a node to current sensor node by copying the node information.
Side-Channel attack	Attacker reads the data and get all the information from sensor node.
Hardware Trojans attack	By maliciously modifying an integrated circuit, the attacker can exploit its functionality to access data executed on that circuit.
Battery Draining attack	Due to size constraints, nodes generally consume small batteries having limited energy volume. This make battery-draining attacks a very powerful attack.
Corrupted/Malicious Node	The key goal of corrupting nodes is to gain unauthorized access and modify the data.
Unauthorized access	Unauthorized access of CPS devices.
Denial-of-Service (DoS)-Jamming	Interference transmission (like jamming) of a radio signal that interferes with the radio frequencies used by sensor node.
Denial-of-Service (DoS)-Laser light	Interference transmission through laser lights on Camera.
Diversion attack	In this type of attack, the attacker divert the data to unknown place without having knowledge to actual user.
Theft of devices	Theft of any CPS devices.
Failure or Disruption of the Power Supply	Besides failures or disruptions of the power supply can also harm the CPS operation.
Hardware manipulation	Hardware manipulation can have any form of targeted objects in an unnoticed way.
Failure/Malfunction of devices	The failure or malfunction of device can lead to a failure of the entire IT operation.
Natural disaster	With natural disasters natural variations, which have a devastating impact on infrastructures, e.g., floods, hurricanes, tornadoes, volcanic eruptions, earthquakes, unfavourable climate conditions and other geologic processes.
Fire	Uncontrolled fire can damage gateways.
Explosion	Explosions can damage sensors/actuators or camera.
Damage/Destruction	Damage or Destruction of gateways by intruders.
Vandalism	Deliberated damage/destruction for no reason.
Terrorism	Attacks by terroristic organizations.

Technique: Following techniques are useful for analyzing the threats:

Misuse case: A misuse case is considered to be a very useful technique to analyze the threats in each layer. The possible threats can easily be analysed through misuse case in the CPS components.

Questionnaire: Questionnaire can also be considered a useful technique by gathering the useful data from users. A set of fixed questions is given to users and which they return back for further analysis. This technique is found to be fast and ranges an enormous user group, which means threats can be considered thoroughly.

Output: Once we analyse the threats from the input list, then it becomes easy to finalize and the threats to be included in the output.

A4: Identify Secure Network Communication

Communication is a shared functionality of nearly all elements in a CPS. It also at times, poses the greatest vulnerability to outside threats. In the requirements engineering phase, we predetermine the most suitable methods of network communication. The purpose of this activity is to identify a secure network communication protocol. Wireless sensor network devices need to be properly authenticated in the network domain. It is important to deploy standard security protocols like Transport Layer Security (TLS), Datagram Transport Layer Security (DTLS), Internet Protocol Security (IPsec) and Secure/Multipurpose Internet Mail Extension (S/MIME). This performs the communication protocol secure for wireless sensor networks.

Based on the security analysis and in the preceding activities, the identified security goals, the security team selects a secure network communication protocol for the participants in the CPS network. Due to the lack of support for CPS-specific aspects of the system by most secure network communication protocols, the security team might find shortcomings of candidate protocols in a number of aspects. In this case, the team documents required adaptations in order to tailor the protocol to the system's requirements in a later phase of the development process. Secure network communication without any security mechanisms would be impossible. That's why the more security mechanisms are installed, the higher is the security of the whole communication. Firewalls, or especially packet-filtering firewalls in networks, allow

to filter incoming and outgoing network packets. The security policy describes which Internet Protocol (IP) addresses or ports are undesirable and the firewall blocks every incoming or outgoing packet with this information in the header [199].

Next to firewalls, Network Intrusion Detection Systems (NIDS) gain are most important. They monitor the network traffic and cause alarm, if suspicious activities are discovered. The flaw of these systems is obvious: they merely notice the user, but do not make any corrective measures to combat the problem. Therefore, Network Intrusion Prevention Systems (NIPS) can be installed, which detect threats and reject malicious traffic without user interaction.

Since communication over a network is non-personal, the system must be able to verify the user (authentication). Furthermore, it needs to be checked, whether the user has some special privileges and is allowed to use the accessed parts of a system (authorization). Access control guarantees both, authentication and authorization and is therefore a very important security mechanism [200].

Depending on the kind of communication, different protocols can be used. Since wireless sensors introduce severe resource constraints due to lack of storage and power, classic security principles cannot be implemented to wireless sensor networks easily e.g. key management, intrusion Resilience [201]. Furthermore, the communication in wireless sensor networks are unreliable, which means that packets get damaged or dropped frequently. Since different protocols direct different characteristics, the choice of the protocols can improve the security significantly [202].

The goal of this activity is to identify secure network communication protocols that enable communication in the CPS network and simultaneously secure the communication links between network participants. The analyst team selects the communication protocols from the checklist of table 5.7, and where required, may add a protocol not mentioned on the checklist.

Input: A checklist of network communication is listed in input [203] [202].

Table 5.7 Checklist of Secure Network Communication

Application Layer	
HTTPS	Hyper Text Transfer Protocol Secure (HTTPS) is the secure communications for the website and are encrypted.
SET	Secure Electronic Transaction (SET) is a system to ensure the security of financial transactions on the Internet.
PGP	Pretty Good Privacy (PGP) makes all communication secure by using encryption. Its use digital signature to verify the authenticity of a document or file.
S/MIME	S/MIME is a Secure/Multipurpose Internet Mail Extension. S/MIME is a technical specification of communication protocols which defines the transfer of multimedia data containing image, text, audio, video and other documents.
KERBEROS	Kerberos is a computer network authentication protocol that allow nodes communicating over a non-secure network to prove their identity in a secure manner.
Transport Layer	
SSL	SSL (Secure Sockets Layer) is the standard security technology used to create an encrypted connection between a browser and a Web server.
TLS	Websites uses Transport Layer Security (TLS) to secure all communications between server and browser.
DTLS	Datagram Transport Layer Security (DTLS) is a networking protocol develop to secure data confidentiality and to prevent tampering and eavesdropping.
Network Layer	
IEEE 802.3	IEEE 802.3 is a standard specification for Ethernet, a process of physical communication in a local area network.
IPsec	Internet Protocol Security (IPsec) is a set of protocols that deliver security for Internet Protocol.
VPN	A Virtual Private Network (VPN) is a secure and encrypted connection for the internet.
Datalink Layer	
PPP	The Point-to-Point Protocol (PPP) is a data connection protocol to which two nodes are directly connected.
RADIUS	The Remote Authentication Dial-Up User Service (RADIUS) is a network protocol. It is designed to authenticate remote users to a dial-up server.

TACACS+	TACACS+ is a Terminal Access Controller Access Control Server Plus is a security protocol that provide centralized authentication for users who want to gain access to the network.
RTP	Real-time Transfer Protocol (RTP) to manage the real-time transmission.
RTSP	Real-time Streaming Protocol (RTSP) is used for creating and controlling media sessions.

Technique: The facilitated meeting session is suitable technique to select the secure network communication. From the input, the detailed analysis and comparison can help to finalize the network communication.

Output: The final list of secure network communication can be generated in the output.

A5: Identify Hardware Endpoint

Failure of hardware endpoint will lead to disruption of CPS operation, which not only interrupt the operation but also lead to vulnerability and open avenues for attacker to threat any CPS devices. Therefore, it is recommended to use only authenticated hardware endpoint. This activity involves the identification of supporting hardware that may include a sensor, machine, router, reader, point-of-sale terminal, server and smart devices. The electronic devices must support other communication methods or channels. Hardware failures may also occur due to design and manufacturing errors or because the hardware has reached the end of its natural life. Operational failure results from a simple fact; human operators make mistakes because of hardware design. For instance, unclear warnings, e.g. a green warning light instead of a red one, or two buttons of radically opposite functionality situated right next to each other, or wrongly shaped sensor probes that make correct measurements in the given environment difficult for the human operator, and so on. This is probably the most common reason of system failures in socio-technical systems today [204].

Therefore, it is recommended to use only authenticated vendor, so trust can be built easily when using good vendor who has good reputation in the market. Security failures can be minimized when having an authentic vendor, preferably a large, well-

established and reputable supplier that is not only known for its product quality, but also has a constant market presence, so as to ensure future availability of hardware supply. Also, it is important to keep note of vendor proprietary parts when deciding on a hardware vendor, as this has the potential to limit the possibilities and reliability of future purchases from elsewhere. Hence, vendor have direct relation with sensor, machine and gateways.

Since for many systems, it is a requirement to run 24 hours, sensors often need to have redundant backups. If an environmental disaster like an earthquake damages or destroys some of them, redundant sensors pitch in – though it must also be noted that the possibility of the disaster disabling the redundant sensors as well cannot be discounted. Consequently, redundancy is not very cost efficient since redundant hardware is not used, if everything works fine. Nonetheless, it is one of the most important security mechanisms for hardware endpoint. While redundancy works fine against environmental circumstances, it does not fare well in the face of deliberate attacks by other human beings, who would likely even destroy the redundant backups, unless the backups are in separate, more protected locations. To be prepared against such kinds of attacks, classic security mechanisms like fences or cameras can be installed. Depending on the cyber-physical systems used, they can ensure a certain level of security, since attackers may not be able to reach the hardware endpoint at all.

Hardware endpoint are required to control the physical process of a CPS. The security team needs to identify the required types and quantity of hardware endpoint as well as necessary features in terms of processing power, memory, energy supply, etc. Furthermore, when installing and configuring the hardware devices, it is recommended to follow IEC 61850 standard [205]. From the checklist of hardware endpoint as shown in the table 5.8, the analyst selects the important hardware endpoint. The result of this activity is a concrete list of sensor, actuator and network devices.

Input: In input checklist, we listed all possible hardware endpoint of CPS [1] [193] [206].

Table 5.8 Checklist of Hardware Endpoint

Computer	Programmable electronic device.
Server	A place to store information.
Sensor	To perceive events in the environment and send the information to gateways.
Actuator	A device to operate the components.
Camera	Cameras for recording critical sections.
Biometric devices	A security identification devices.
Smart meter	A measurement device.
Router	A networking device that pass data packets.
Point-of-Sale terminal	A computerized device.
Gateway	Serves as the connection point between the physical environment and controller.
PLC	An industrial computer control system that constantly monitors the devices.
Controller	A control system that manages and regulates other systems.
Transformer	A machine to manage the voltage.
Monitor	Hardware to show the graphical user interface.
Arduino	An electronics platform or board that software is uses to program it.
Cable	A wire to use for transmitting electricity or telecommunication signals.
Monitor	Hardware to show the graphical user interface.
Fences	Fences to prevent intruders from getting access to physical objects.
Range extender	A device that extends the range of communications signal.
Equipment	Set of different devices and tools.

Technique: The facilitated meeting session is very useful technique, especially when there is an issue concerning which vendor are more appropriate in terms of security. Furthermore, this technique is useful to analyze and finalize the hardware endpoint.

Output: After applying the above technique, the final list of hardware endpoint is generated.

A6: Identify Sensor Types and Communication Medium

The sensor types and communication is one of the critical properties of the CPS. For most cases, sensors can be located in remote locations away from the main processing unit / controller of the CPS. In such cases, ensuring secure communication to and from the sensors becomes extremely important. For these purposes, it is most convenient to

make use of wireless sensor communication, and as a result, it is seen to be more commonly used than wired connections, though wired connections are still used where they are more suitable for data security.

Wireless Sensor Networks (WSN) are networking structures comprised of many small and low-cost sensor nodes, which have limited computational power and energy supply. The main goal of these networks is sensing, actuating and sending of environmental information to a data sink, which then processes it. The biggest advantage of WSN—their autonomous and unattended operation—in turn also opens up many attack possibilities, e.g., tampering, physical manipulation and node compromise, due to the unavoidable disappearance of a security perimeter.

Modern sensors offer a wide area of challenges, and not only because they get continuously smaller and resources like power, memory or computational capacities are limited. As cyber-physical systems communicate in real-time, sensor nodes must guarantee data 'freshness' [207]. Furthermore, cyber-physical systems may interact in several, uncertain environments with different characteristics such as temperature / pressure or other similar conditions. To perform well in every circumstance, sensors need to be very robust, despite their small sizes, and need to include some security mechanisms [208].

Various wireless technologies, such as Radio Frequency (RF), Bluetooth, and Zigbee, have been applied to sensor communications. Sensors and actuators are devices that communicate with the external environment. The sensors generate data regarding the object with which they are in contact. These data are received through gateways and pass to the main controller for further processing. These sensors use Machine to Machine (M2M) protocols for communications. Since different sensors depict different mediums of protocol, we have proposed to identify all such protocols during sensor analysis.

Furthermore, the sensor communication medium may have one or more security goals and / or threats, such as, a threat to availability, similar to a Denial-of-Service (DoS) attack or Man-In-the-Middle (MIM) attack. A sensor communication protocol enables communication in sensor networks of the system, i.e. between the sensors, actuators and gateways. The analyst team would select the sensor type and Sensor Communication Medium (SCM) according to the nature of CPS project.

Input: A checklist of sensor types and communication medium is listed in input as shown in table 5.9 [209] [210] [211].

Table 5.9 Checklist of Sensor Types and Communication Medium

Temperature Sensor	To measure the temperature in an environment.
Ultrasonic Sensor	To detects the presence/absence and calculate the distance of an object.
Pressure Sensor	To sense the pressure from the environment and converts into an electric signal.
Smoke Sensor	To sense the smoke from the environment.
IR (Infrared) Sensor	To sense certain aspects of its surroundings.
Motion Detection Sensor	To detect the physical movement from the environment.
Image Sensor	To detect and transfer the information to create an image.
Optical Sensor	To measures the physical capacity of light rays.
LTE 4G	4th Generation, mobile data communications technology
IEEE 802.3 (Ethernet)	It is used for computer networking and general data communications.
Wifi	Link to portable device through internet connection.
ZigBee	ZigBee is an open global standard designed precisely for use in M2M networks.
Near-field communication (NFC)	NFC uses as a communication protocols that enable two devices to communicate with short range only.
Wi Max (IEEE 802.16)	Wi Max technology allows data to transfer at a rate of 30-40 megabits per second.
Radio Frequency (RF)	RF is a form of electromagnetic transmission that use in wireless communication. The range of RF is from 3kHz to 300GHz.
Infrared	Infrared is in the form of electromagnetic radiations. It uses for short-range communications.
Bluetooth	Bluetooth is used to transfer data for short distances. This technology is often used in small devices.
Z-Wave	It is a wireless communications protocol that used mainly for home automation.
6LowPan	6LoWPAN uses for the smallest devices having limited processing capability to communicate wirelessly through IP.

Technique: The facilitating meeting session is useful technique to compare and analyze the sensor communication medium. The analyst team and stakeholders can analyze which sensor type and communication medium are useful according to the nature of the project.

Output: Finally, the analyst team and stakeholders can finalize the sensor types and communication medium and generated in the output list.

A7: Perform Risk Assessment

The purpose of this activity is to evaluate the level of risk to a CPS, where risk can be seen as the value of the expected loss to an asset due to a given threat. In the risk assessment activity, for each asset, its associated threats and vulnerabilities are estimated. Here, we have extended the Risk Assessment methodology proposed by National Institute of Standard Technology (NIST) [212] for suitable use with CPS. There are few other risk assessments methods [179] [213] [214] that can be used, depending on the expertise of the analyst team, but we have selected this method as it is both well-known and concise. Risk assessment can be supported by additional information sources, for instance historical data, statistics and experiences from experts and stakeholders. The results of this activity serve as a basis for deciding which security risks shall be addressed by implementing security mechanisms and which security risks are regarded as inconsequential and hence as tolerable [179]. The basis of every potential risk is always a certain threat, for which, the associated risks can be prioritized on the basis of the 'likelihood' of occurrence and the 'impact' of the threat, given that the event occurs. Once the risks are prioritized, response strategies which deal with reducing the impact of the risk can be implemented.

Risk prevention is the optimal solution, but also the most difficult to achieve. The intention here is to be able to counter the negative effects impacting the system, a process which may involve applying major changes to the project design. Since this is a rather costly procedure, strategies such as removing the affected system component

could be applied, too. Obviously, this solution limits system functionality and it may not always be clear whether the impact of the threat is high enough to warrant removal of this component.

While risk prevention needs to be implemented in the beginning of the development process, risk mitigation can also be applied later. The history of software development has been shown that it is not possible to be aware of every threat to a project. If the development process has nearly finished and an unexpected threat emerges which would necessitate dramatic changes to the system design, risk mitigation through countermeasures, which reduce or ultimately eliminate the risk, can be implemented [215].

Likelihood:

The likelihood of a threat indicates the probability of a successful attack. The values for likelihood are divided in five categories – very high, high, medium, low and very low. The categories are outlined as follows:

- *Very High:* The attacker has very high motivation and sufficient capabilities for performing an attack; the system lacks security mechanisms.
- *High:* The attacker has high motivation and sufficient capabilities for performing an attack; the system has insufficient controls to combat the threat.
- *Medium:* The attacker has sufficient motivation and capabilities to perform an attack; the attack might be prevented by security mechanisms that are in action.
- *Low:* the attacker has low motivation or lacks capabilities, or the system possess sufficient controls to combat the threat.
- *Very Low:* The attacker lacks motivation and capabilities, or the system possess very strong security mechanisms.

Impact: Impact of a threat indicates the effect on an associated asset of a threat if it is successful. This effect could be in the form of monetary cost, time, effort, or reputation. It should be noted that different stakeholders may not all look at different assets with the same frame of reference. To temper this difference in perspective, considering a third opinion from an expert can help greatly in obtaining a relatively objective evaluation of the impact of a threat.

- *Very High:* Very severe impact; puts an asset at a very high loss – countermeasures are indispensable.
- *High:* Severe impact; puts an asset at a high loss – countermeasures are indispensable.
- *Medium:* Average impact; consequences of an attack harm the system, its assets and interfere the correct functioning of the system – countermeasures are required to reduce impact or prevent the attack.
- *Low:* Consequences are at a tolerable and do not interfere in an asset – it should be determined if countermeasures are required.
- *Very Low:* Consequences are very low to negligible and do not interfere in an asset – it should be determined if countermeasures are required.

Table 5.10 below shows how the risk ratings are evaluated based on inputs from the likelihood and impact categories for each threat. The threat likelihood for any given asset ranges from 1 to 5, with 1 being very low, and 5 being very high. Similarly, the impact of a threat on a given asset ranges from 10 to 50, with 10 being very low and 50 being very high.

Table 5.10 CPS Risk Matrix

Threat Likelihood	Impact				
	Very High (50)	*High (40)*	*Medium (30)*	*Low (20)*	*Very Low (10)*
Very High (5)	**Very High** 5x50= 250	**Very High** 5x40=200	**High** 5x30=150	**Medium** 5x20=100	**Low** 5x10=50
High (4)	**Very High** 4x50= 200	**High** 4x40=160	**Medium** 4x30=120	**Low** 4x20=80	**Very Low** 4x10=40
Medium (3)	**High** 3x50= 150	**Medium** 3x40=120	**Low** 3x30=90	**Low** 3x20=60	**Very Low** 3x10=30
Low (2)	**Medium** 2x50= 100	**Low** 2x40=80	**Low** 2x30=60	**Very Low** 2x20=40	**Very Low** 2x10=20
Very Low (1)	**Low** 1x50= 50	**Very Low** 1x40=40	**Very Low** 1x30=30	**Very Low** 1x20=20	**Very Low** 1x10=10

The likelihood and impact values are then used to calculate the risk. Feedback from the clients themselves as well as expert opinions may be utilized to determine the likelihood and impact on each asset subjected to the risk. The likelihood and impact values are then multiplied to calculate each risk based on the risk scale. It must be said that estimating the probability of occurrence may be difficult in practice. In such a case, only the relative order of the probability estimates become relevant, leaving us free to estimate probabilities using a relative risk scale [216]. The risk scale is shown in table 5.11.

Table 5.11 Risk Scale

Risk value	Risk	Risk Description
200 to 250	Very High	Organization is likely to face very severe and repeated losses. Strong preventive controls are absolutely necessary.
150 to199	High	Organization may face sizable or repeated losses. A certain degree of preventive control is required.
100 to 149	Medium	Organization may suffer financially but limited liability or loss of reputation. Corrective procedures must be in place.
50 to 99	Low	Organization may face limited financial loss or loss of reputation. Corrective procedures may be installed if feasible.
1 to 49	Very Low	Organization is not likely to face loss of finance or reputation. Mitigating action not required.

The risk scale, with its ratings of Very High to Very Low represents the degree or level of risks to which the system might be exposed if a given threat were exercised.

Input: The output of assets and threats diagnosis become the input of risk assessment. Usually, risk for assets can be assessed given a threat's likelihood and impact.

Technique: CPS risk matrix and misuse case.

The analyst team attempts to identify and describe action sequences through techniques like the misuse case, detailing how threats exploit vulnerabilities, leading to undesirable events which may cause damage or other form of loss to one or more assets. This identification process is led by the risk analysis leader, using the most valuable assets, e.g. by posing relevant questions to the risk analysis team. The use of misuse cases facilitates understanding and communication between the participants. Questionnaires, checklists and other tools may be adopted to help support the process of threat and vulnerability identification.

One of the most critical elements that may influence security requirements is financial and logistical feasibility of implementing any given security feature. For this, it becomes very important to conduct a cost-benefit analysis when implementing

security requirements, particularly for assets faced with medium to very low risk. A cost-benefit analysis represents an estimation and comparison of the relative cost and value of different proposed controls.

For this purpose, estimates of the impact of implementing the new control, the impact of not implementing any control, the cost of implementing the new control are assessed. If the implementation reduces the risk at an acceptable level and is cost effective, the measure should be implemented. Otherwise, a more effective or less expensive measure should be sought. In the case where the control reduces the risk level more than is necessary, a simpler and more cost effective alternative should be chosen [217].

Output: risk.

After the analysis of risk matrix, the analyst team generates the risk. The highest values need to be given the most attention to elicit security requirements.

A8: Elicit Security Requirements

The purpose of this activity is to elicit, analyze, and specify the security requirements. Precise and unambiguous requirements are organized and written down. The results identified from the preceding activities, i.e., assets, threats, security goals, secure network communication, hardware endpoint, sensor communication medium, and the result from the risk assessment, are incorporated to determine the security requirements. The output of all activities further analysis through misuse case to elicit security requirements. The risk assessment provides means for evaluating priorities for addressing security risks. The result of this activity is a complete set of documented security requirements for the analyzed CPS.

Once a list of security requirements is generated, it is further recommended to validate the specified security requirements. The validation can either be performed by the security team or assigned to an inspection team. Security requirements are validated

in terms of consistency and correctness. An essential aspect in validating security requirements for CPS is to ensure that the requirements address the correct system context and all security-critical aspects of the system. This means that security requirements additionally need to be validated in terms of CPS security.

After the completion of all preceding activities, security requirements are elicited.

Input: The output of assets, security goals, threats, secure network and sensor communication, hardware endpoint and risk result will become input of this activity to elicit security requirements.

Technique: Facilitating meeting session and misuse case.

Facilitating meeting session: The analyst team and stakeholders analyses the identified assets, security goals, threats, secure network and sensor communication, hardware endpoint and risk. It is recommended all analyst teams and stakeholders should gather and have a discussion to elicit security requirements. All assets, security goals, threats, secure network and sensor communication, hardware endpoint and risks of the system are subjected to detailed analysis by the stakeholders through misuse case. The SRE Tool is designed to export the results from these seven activities onto a single file which makes the process of security requirements elicitation significantly easier. Keeping in view these results from the previous activities, we combine them together to form security requirements. This can be done by individually considering different assets, together with various aspects of security goals and threats, and discussing their security at length with the key stakeholders. It should be noted here that for problems in the field of requirements engineering, meeting session through scenarios like use case and misuse case are considered to be the most suitable method to elicit (security) requirements [218] [123] [183] [219] [64]. Hence, misuse case facilitates understanding and communication between security team and stakeholders. Furthermore, an industrial survey [220] also reports that scenarios are suitable technique to determine and validate requirements that make

them agreed and consistent [183]. Use cases are popular tools for eliciting functional requirements while misuse cases are considered to elicit security requirements.

Output: After applying the above techniques, the analyst team and stakeholders can finalize the security requirements. These requirements need not all be at an equal level of importance. They could be categorized and prioritized into hold points, necessary requirements and recommended security requirements.

5.3.3. Technique Misuse case

A technique in common use today for eliciting, communicating and documenting requirements for a system is the 'use case'. This is a tool that works to elicit functional requirements. However, as in the case of security requirements, it is not tailored to looking for extra-functional requirements [183]. Nevertheless, the idea that the use case was built upon has shown to be a solid one, and use cases have proven useful in general for the elicitation of, communication about and documentation of requirements [221] [222] [223]. The main idea here is to describe some function that the system is meant to be capable of performing. Seeing that security requirements are an added feature to the main functionality of the system, and in the absence of threats, have no bearing on the actual working on the system, it can be seen how a 'use case' would be helpful for working with functional requirements, but not so much with extra-functional ones, as is the need in our case [218] [224] .

Therefore, we have chosen to utilize the 'misuse case' technique as proposed by Sindre and Opdahl [183] in our proposed framework to analyse threats, risk assessment and elicit security requirements. A misuse case is the inverse of a use case, i.e., a function that the system should not allow. A use case is defined as a complete sequence of actions that provides the user a higher value. On the other side, a misuse is defined as a complete sequence of actions that leads to losses for the organization or a particular stakeholder [225]. A misuse case supports the security team and analyst to determine

action sequences that accurately describe how threats exploit the vulnerability and lead to unwanted events that can cause damage or other loss to one or more assets.

A 'misactor' is the inverse of an actor, i.e., an actor that one does not want the system to support, an actor who initiates misuse cases as shown in the figure 5.3. Furthermore, we have introduced four iterations:

- **Developing Misuse case:** We start with developing a misuse case. The analyst team and other stakeholders provide the necessary input in this iteration.
- **Audit Misuse case:** Once we have developed the misuse case, then we further analyse the misuse case according to its need. For example, by analysing the attacker's capabilities.
- **Propose change:** In this iteration, if the misuse case proves unsatisfactory, then further changes are to be proposed from the analyst team. For instance, by adding a new threat from the attacker.
- **Plan change:** We plan the revised misuse case and produce the final version of misuse case.

Figure 5.3 depicts a misuse case diagram. A misactor (e.g., Attacker) is an actor or element that forms the basis for initiating misuse cases, either deliberately or unintentionally. An example of a misuse case could be one detailing a node replication attack, which describes a misactor that diverts important data to an unknown database which creates serious consequences as important data is being stolen and going somewhere else. Furthermore, misuse case diagram not only supports to identify the threats but also assist to elicit security requirements because the misuse case has already been proved to determine the security requirements for software [225]. Similarly, we can infer from the misuse case diagram a security requirement too, e.g. "The sensor shall not divert the data to unknown server". Therefore, we also utilize this technique in our proposed framework to elicit security requirements for CPS. Hence, a misuse case is considered to be a very useful technique to identify the

possible methods of attack on system assets. This will help to identify threats in each layer.

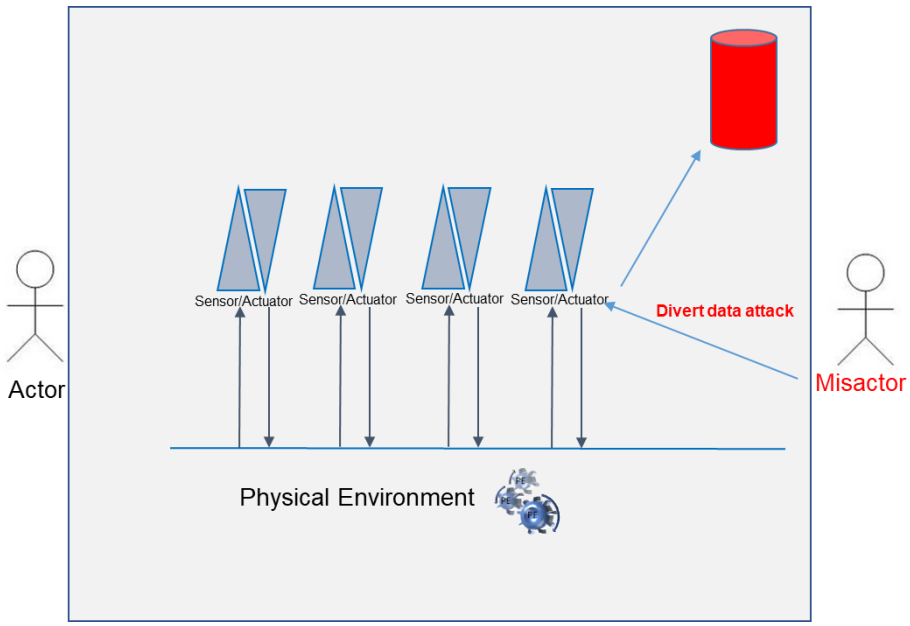

Figure 5.3 Misuse case

5.4 CPS Tool Implementation

This tool is developed in JAVA with a fairly simple Graphical User Interface (GUI), where the user just selects options related to each activity and at the end, the tool generates an SRE document which could then be further extended to complete security requirements for cyber-physical systems.

In the following, we have given step by step illustrations for the said tool:

5.4.1 Main Screen for creating a Project

To start requirement elicitation on the tool, a user needs to start a "New Project" or if a project is already created, then it could be reopened with the button "Open project".

5.4.2 Main Screen to define Activity and Technique

When a user has already created a project then user could add activities which would in turn provide a road map for security requirements. All the activities are loosely

coupled and user could start from any activity but we recommend that the given sequence should be followed for better results. If it is required, a user could do multiple iterations on these activities to cover all security requirements of the system.

5.4.3 List of defined activities

There are eight activities defined in the tool to collect information to cover security requirements. These activities are straightforward and could be performed by system stakeholders independently or in collaboration with requirement engineers. All the activities available in the tool.

5.4.3.1 Identify Assets

In this activity, all the items which have significance for the related system are given in the form of a checklist. The tool has a list of general items given in the left hand side of the User Interface. A user could select already given "Assets" from the list and could also add new "Assets" by clicking the right mouse button, which are required for the system as shown in figure 5.4.

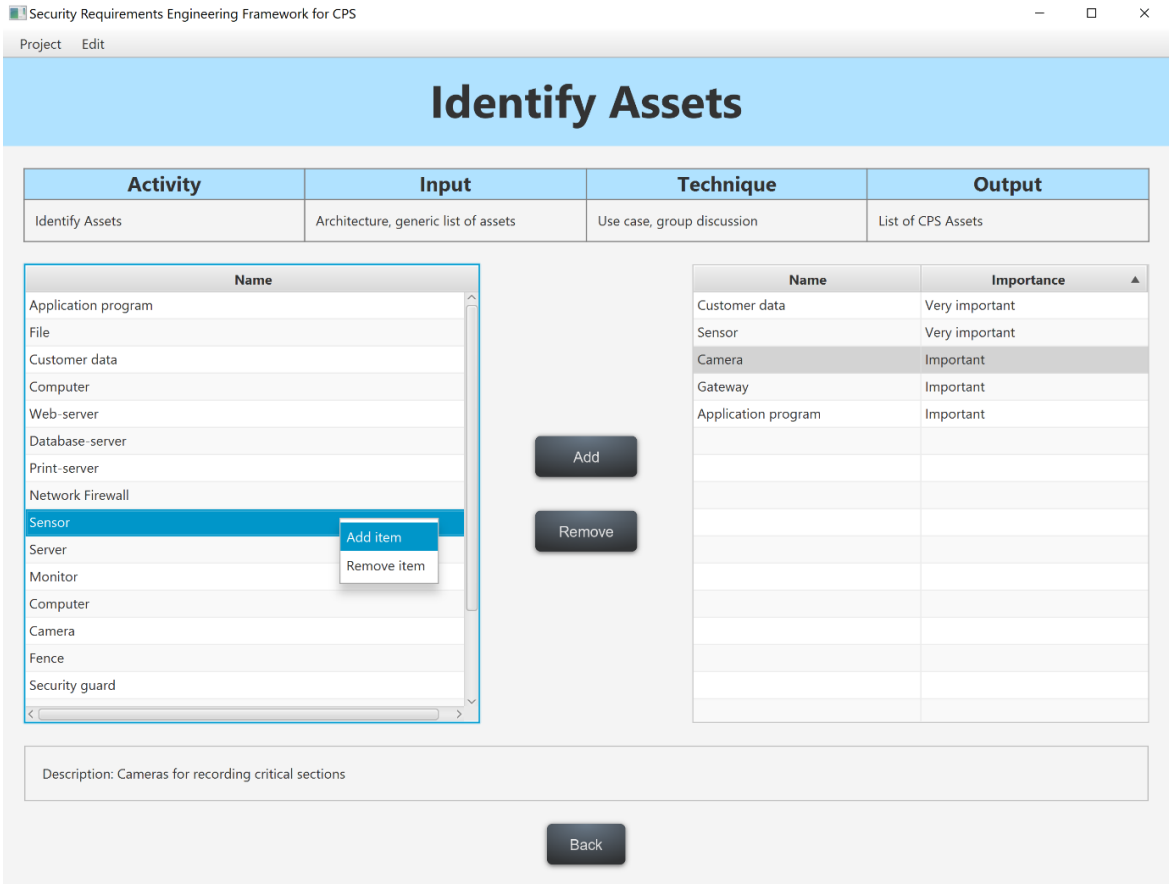

Figure 5.4 Identify Assets

5.4.3.2 Identify Security Goals

In this activity, user select the security goals from the checklist. The tool provides the generic list of security goals. The output of previous activity of assets helps to determine the security goal as the user apply the security goal in every identified assets and this makes easier to finalize the security goals. User can also prioritize the identified security goals according to the system's need and their importance as shown in figure 5.5. However, this feature is optional and not mandatory. By default, the tool is assigned "Important" for every generated output. Priority levels can assign from the following:

- Very Important
- Slightly Important
- Important

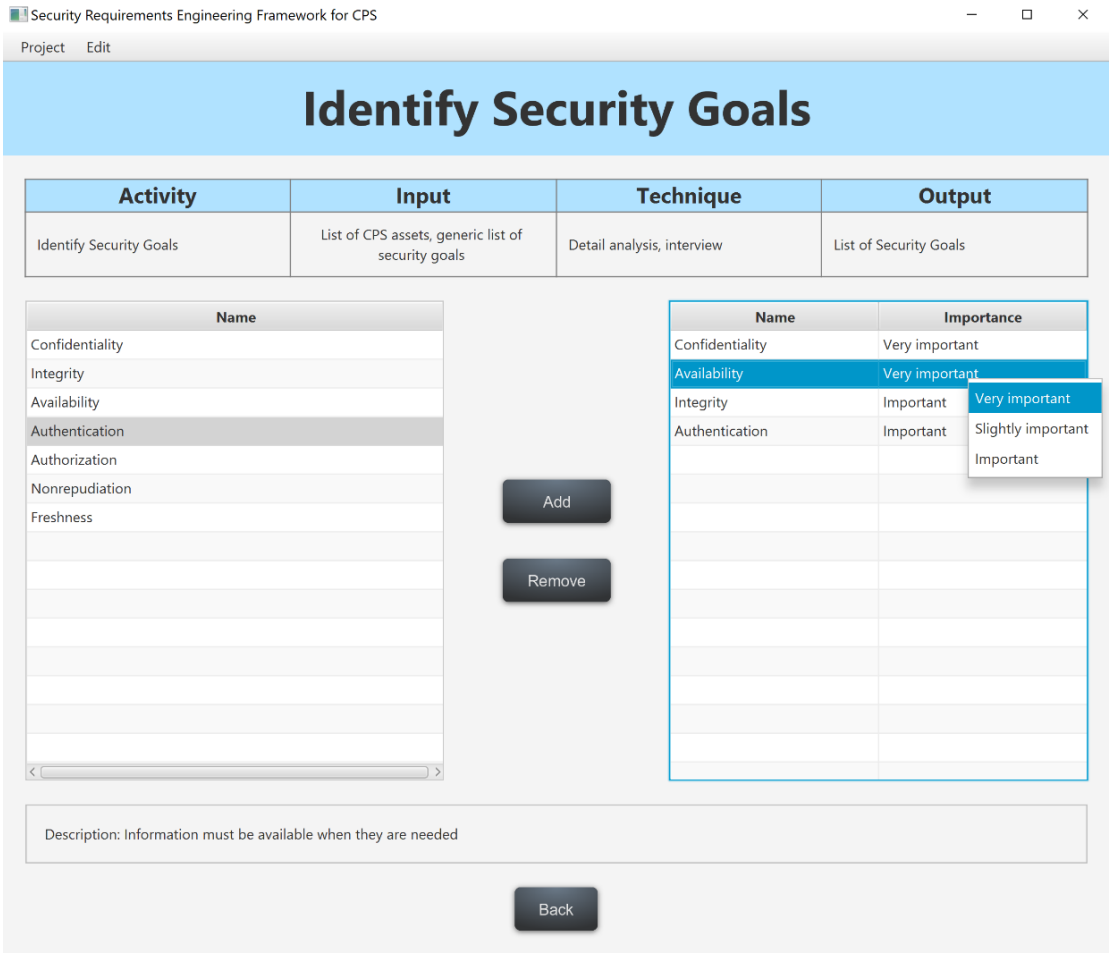

Figure 5.5 Identify Security Goals

5.4.3.3 Identify Threats

Threats related to the system are collected during this activity. Threats could be deliberate or accidental and they could be of different severities. A user selects all relevant threats from the checklist. Threats would be applied in the output of assets and security goals that analyzed by 'misuse case' technique, which makes easier to finalize the threats. Severity should always be appropriately adjusted wherever it is required. An example is shown for threats in the following figure 5.6.

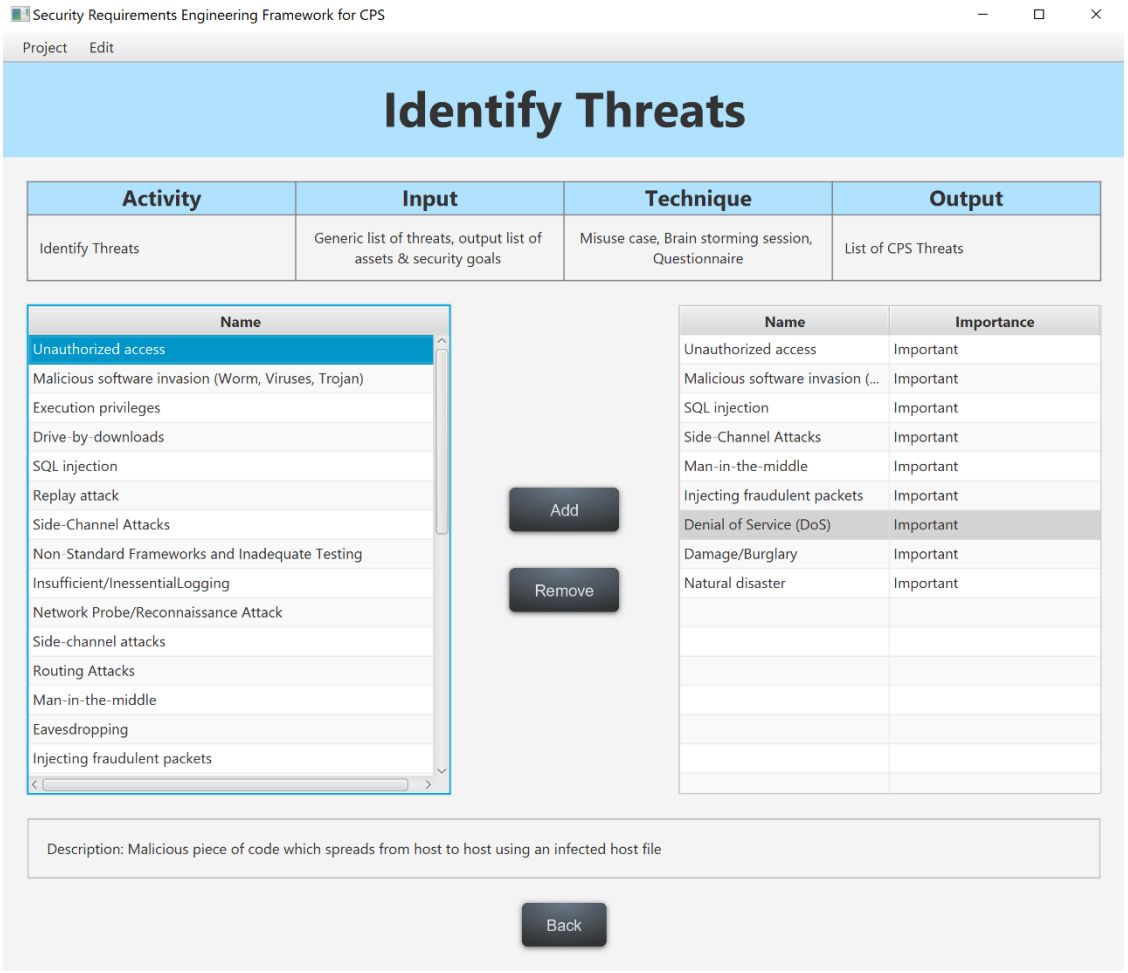

Figure 5.6 Identify Threats

5.4.3.4 Identify Secure Network Communications

As the data needs to be communicated over the network, it is very important that this medium is scrutinized properly. This activity provides this facility where a user could define all network related security matters on the basis of priority as shown in the following figure 5.7. Secure network communications are identified through the technique of analysis and comparison on CPS architecture. The output of this activity will directly help to elicit security requirements.

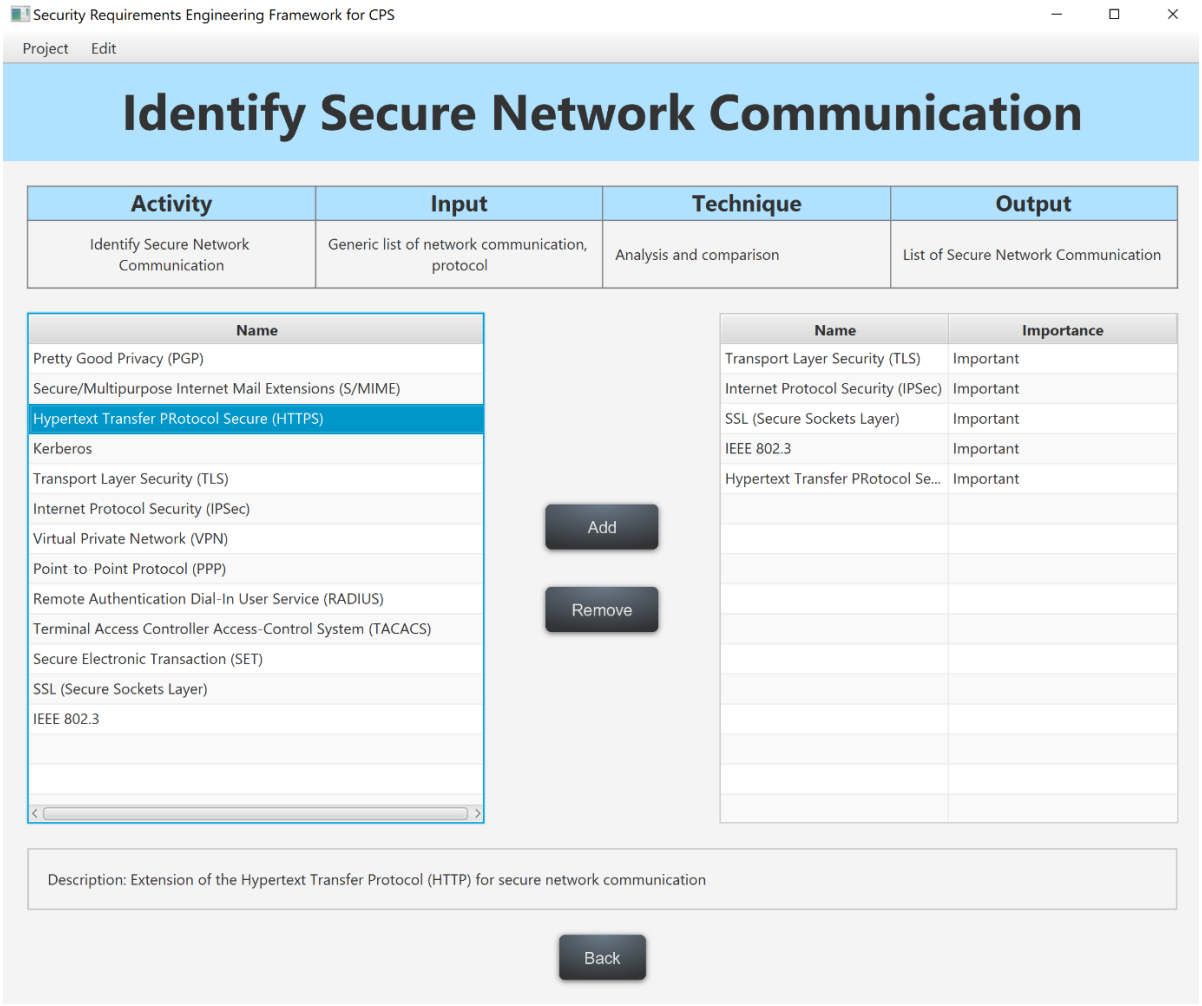

Figure 5.7 Identify Secure Network Communication

5.4.3.5 Identify Hardware Endpoint

Hardware endpoint which is used in the secure components of CPS should also be identified and listed in the tool according to their importance. A general checklist of hardware endpoint is available in the tool which could be also extended according the system requirement. An exemplary checklist of hardware endpoint is given in the figure 5.8.

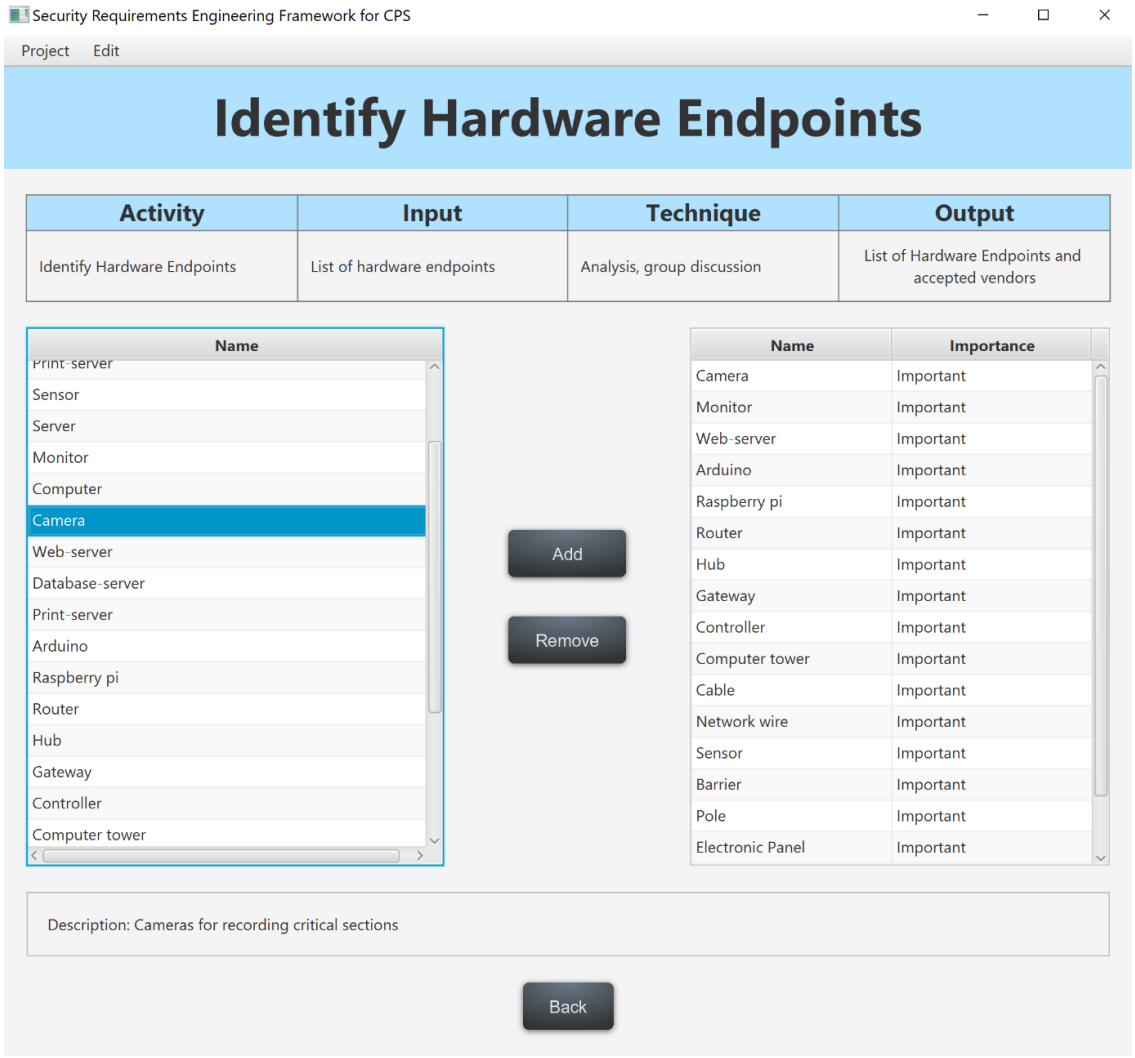

Figure 5.8 Identify Hardware Endpoint

5.4.3.6 Identify Sensor Type and Communication Medium

The communication protocols for each sensor, which are required for the communication in the network, are listed in this activity. In this activity, the analysis and comparison technique helps to identify sensor types and their communication medium. So that required output could be met easily. One could add new communication protocol and prioritize according to the given need as shown in the following figure 5.9.

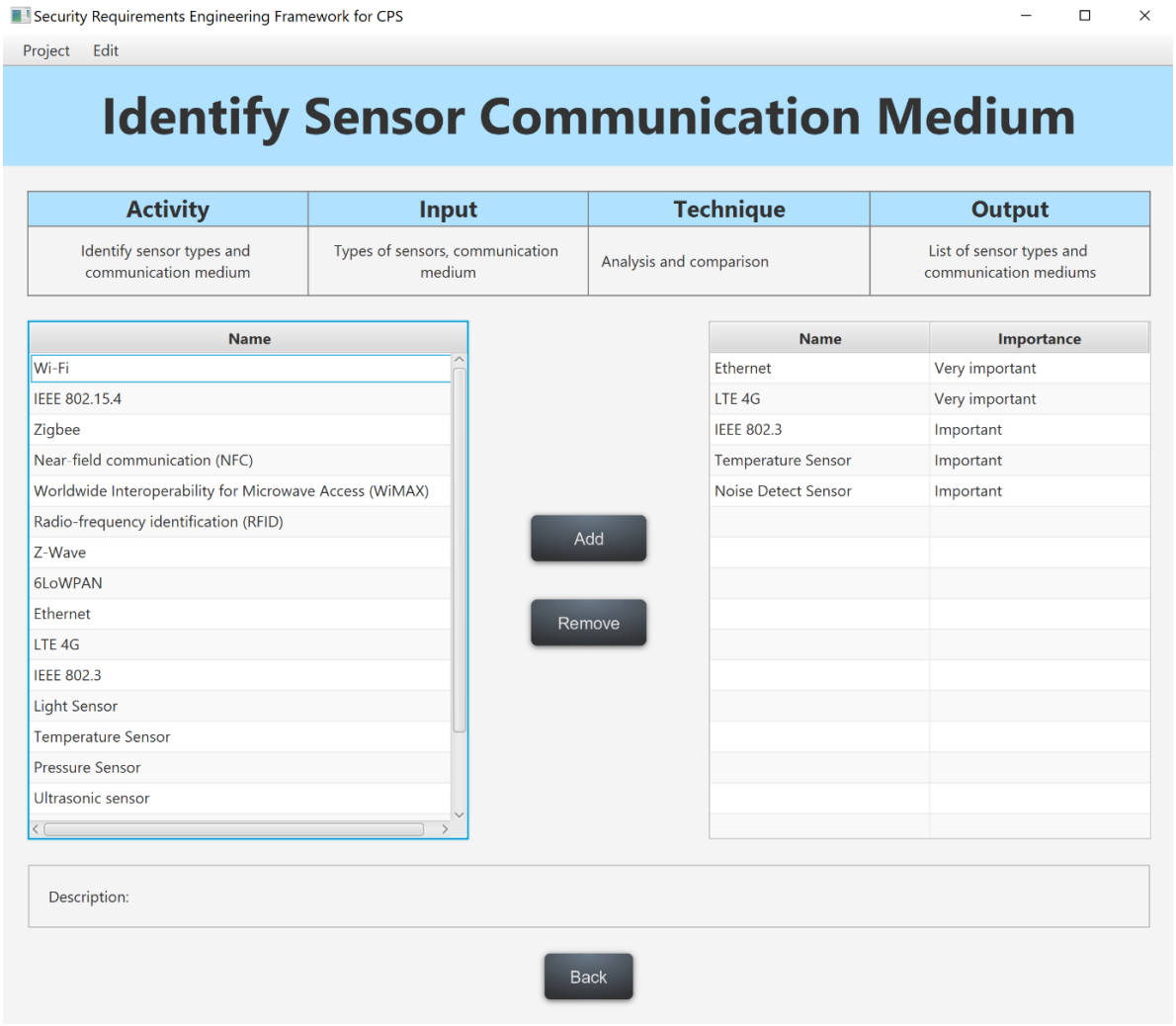

Figure 5.9 Identify Sensor Types and Communication Medium

5.4.3.7 Perform Risk Assessment

This activity performs the risk assessment for CPS. The risk for asset is assessed to threat likelihood and impact. Each asset would be listed in the activity by clicking the right mouse button in the 'Risk ID' field. Security analyst could put threat likelihood and impact value on asset. After consensus the risk list would be generated with the values as shown in the following figure 5.10.

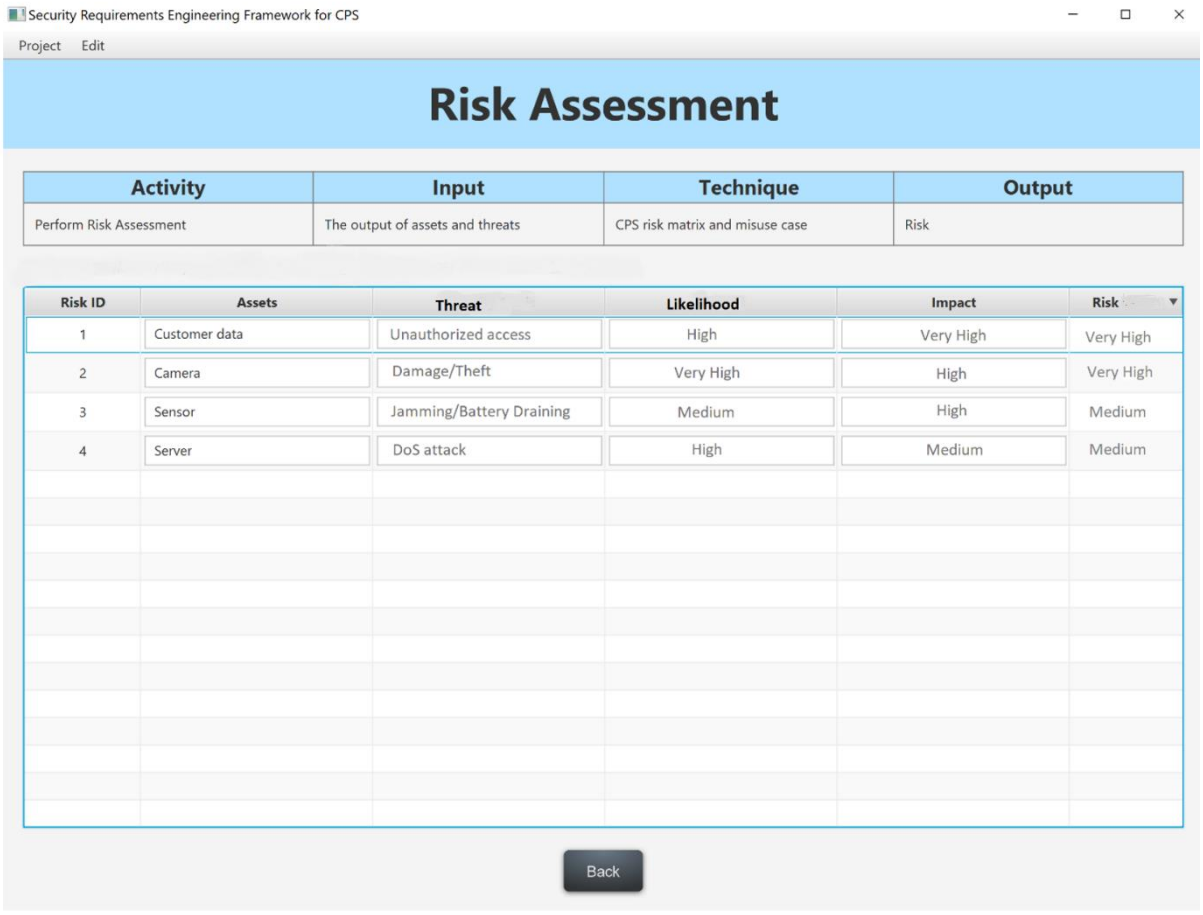

Figure 5.10 Perform Risk Assessment

5.4.3.8 Elicit Security Requirements

The results from all the preceding activities help to elicit security requirements. The outputs of the preceding activities are further analyzed by the security analyst and stakeholders to elicit security requirements. The Tool is designed to export the outputs from these preceding activities onto a single file which makes the process easier to analyse and elicit the security requirements. Finally, this activity lists all the security requirements that help to build secure CPS. Figure 5.11 shows the determine security requirements.

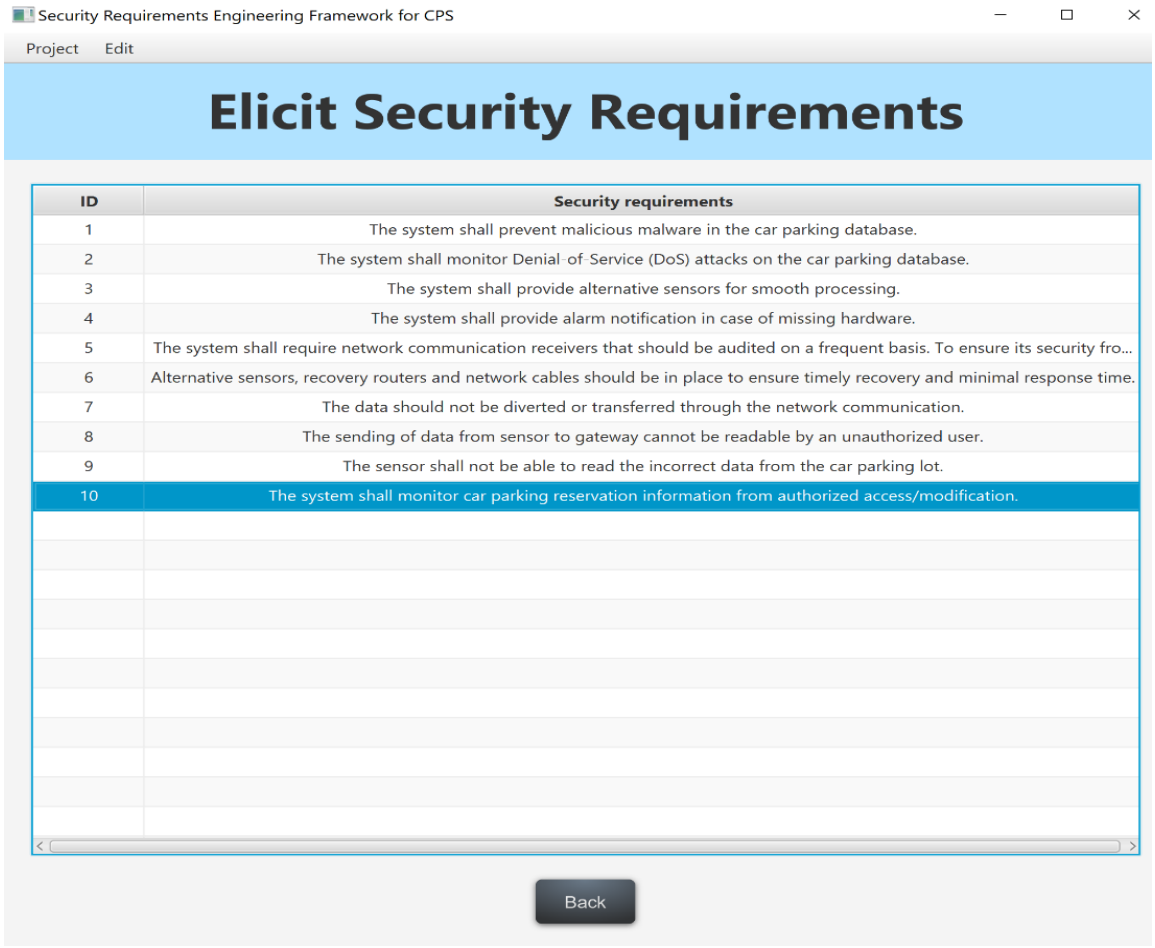

Figure 5.11 Elicit Security Requirements

5.4.4 Developing Use case, Misuse case and Architecture

A tool is also available to create use case, misuse case and architecture. Use case assist to finalize the assets of the system, misuse case helps to analyze the threats and architecture support to analyze the CPS environment. A misuse case of unauthorized access and malicious software invasion is created in the following figure 5.12 with the support of the tool.

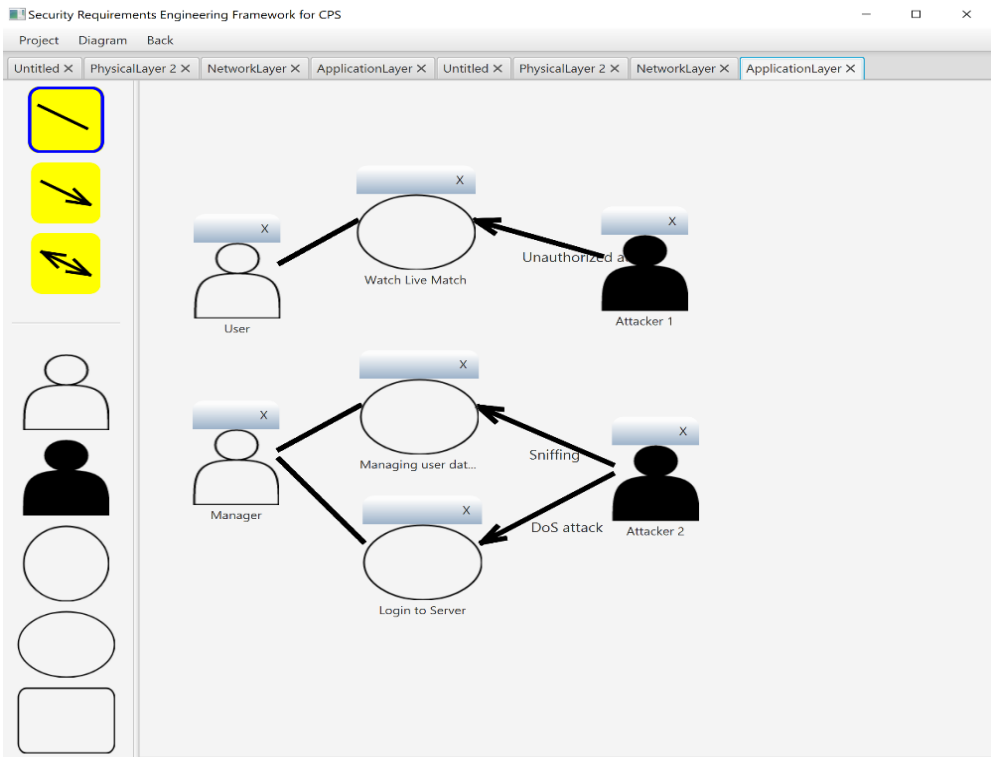

Figure 5.12 Misuse case

5.5 Chapter Summary

Security requirements are a significant part of cyber-physical systems, but there is a lack of processes at present to develop secure systems. Many security requirements methodologies have been proposed, but these are limited only to software, and none supports cyber-physical systems. In this chapter, our main contribution is to provide a comprehensive security requirements engineering framework for cyber-physical systems that can offer complete guidelines for practitioners and researchers to determine security requirements. These activities identify security goals and requirements to prevent and deal with potential consequences of attacks on a CPS. The purpose of this framework is to develop early security concepts in the requirements engineering phase. This quest leads to RE methodologies so that security concerns can be addressed during the early stages of software development. The proposed framework is a systematic approach to incorporate security goals, threats, and risk assessment that are critical to the CPS. We have a set of 8 main activities and

one important technique called misuse case. A misuse case is operated like a use case, just being its converse. The novelty of this contribution is because such an implementation at this scale has not been significantly reported in the literature.

CHAPTER 6

Evaluation of Proposed CPS Framework

In this chapter, we evaluate our security requirements engineering framework for cyber-physical systems by applying it to two case studies. The case studies are conducted with reference to security requirements elicitation with the help of the security requirements engineering Tool we have developed for this purpose. In these case studies, while the cyber-physical systems are considered holistically for security requirements, special attention has been paid to the physical layer of these CPS, given that they are not only more vulnerable to external threats, but also that most existing frameworks have dealt exclusively with the application layer. The first case study involves a smart car parking system, employing a functional prototype consisting of a physical and a software implementation, developed to demonstrate the working of such a smart car parking system. This is described in Section 6.1.

The rest of the chapter is organized as follows: Section 6.2 presents the second case study which was conducted with industrial collaboration from Soccerwatch GmbH. Similarly, we also applied the proposed framework in this real-world scenario. Section 6.3 compares the proposed framework with other SRE frameworks. Section 6.4 summarizes this chapter.

6.1 Case Study 1: Smart Car Parking System (SCPS)

The proposed CPS framework has been applied on a Smart Car Parking System (SCPS). A functional prototype was developed, having both a physical and a software implementation. The website or mobile application facilitates the reservation of available lots in the parking area. The user should register on the website or mobile application and select a payment method for paying the costs of reservations. The user can see the available parking lots and reserve a free lot for a certain time. The parking

lot display in the application is changed according to the information received from the physical layer. The data received from the controller is processed in the web and mobile application. Furthermore, the application controls the physical layer by sending commands to light up a specific LED in case a reservation has been requested by a customer. The most important field in the registration is the car license number provided by the user. This license number will be recognized by camera for allowing the users to use the parking. Figure 6.1 shows the functional prototype; each parking lot is connected to a Raspberry Pi 3, which collects the evaluated sensor data from the Arduino and sends it to the ARTIK cloud. The system maintains the information of the vehicles entering the parking area. The sensor is used to determine the identity of the vehicle. The user interacts with the system through a mobile application. Only registered users are able to use the system.

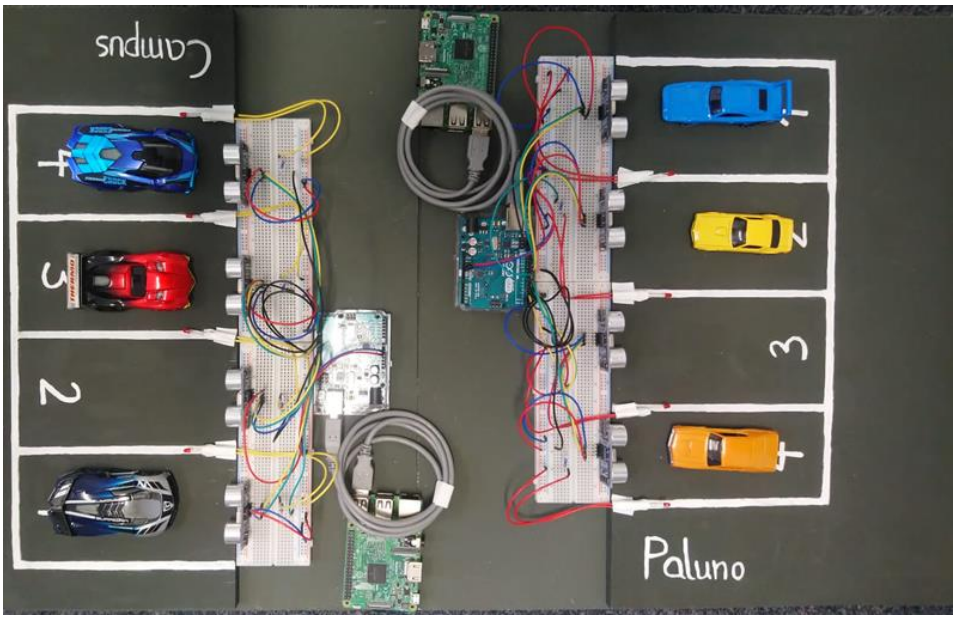

Figure 6.1 Functional Prototype of Smart Car Parking System

The smart car parking system contains equipment and technologies, which are implemented in the physical, network and application layers. Figure 6.2 shows the architecture employed for the system, which combines the functionalities we previously presented in [1] and the developed functional prototype for the smart car parking system as shown in Figure 6.1 above. This supports us to analyse the system

environment as this is first component of our proposed CPS framework and accordingly we design the architecture of smart car parking system. This helps us to understand the functionality of the smart parking lot in the real-world scenario.

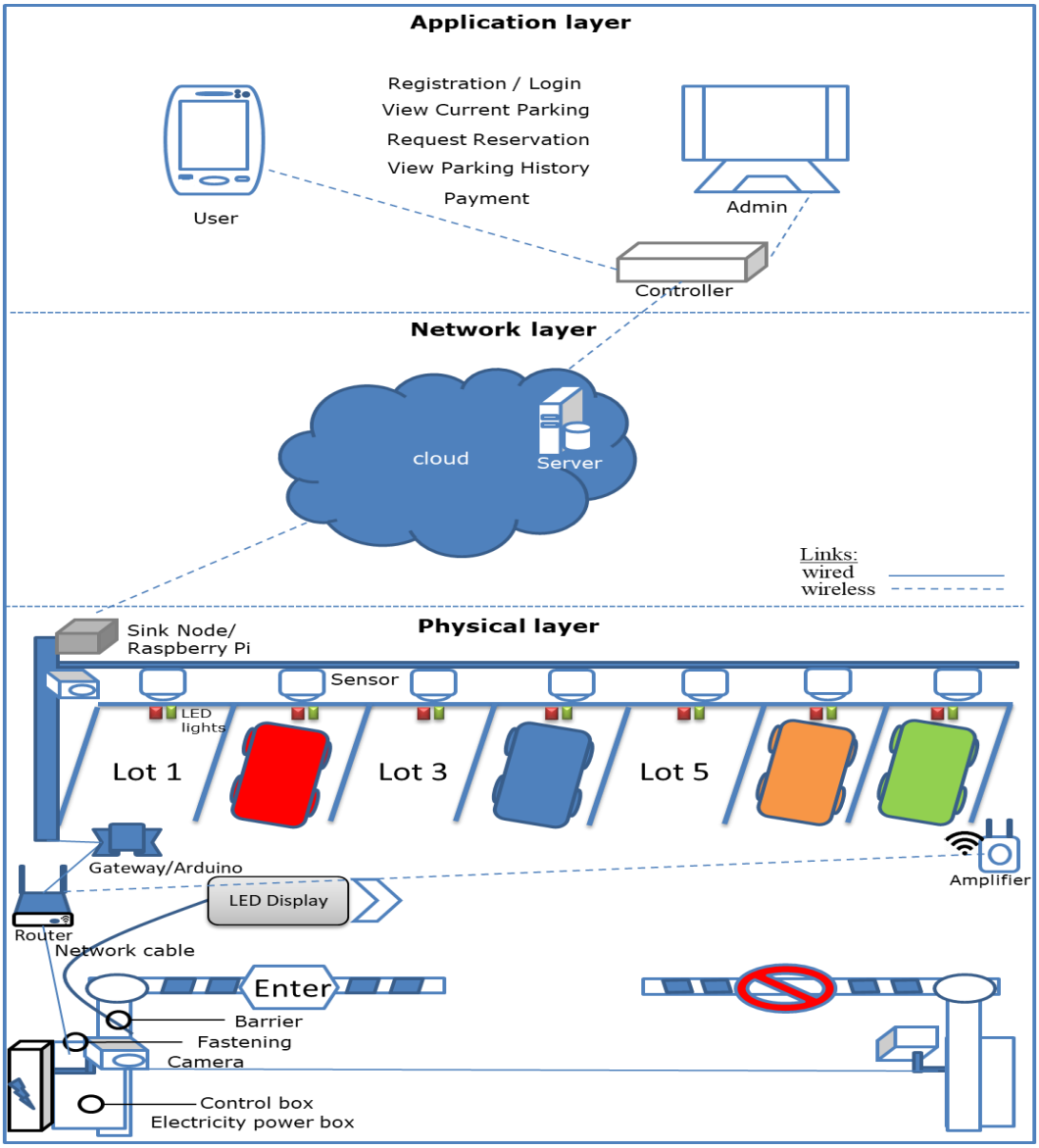

Figure 6.2 Architecture of Smart Car Parking System

Physical Layer:

On the physical layer, the parking system has a set of ultrasonic sensors, which sense or detect a parked car, and LEDs as actuators which indicate the parking status, i.e., whether the parking space is free or occupied / reserved. Two technically identical parking lots were implemented as shown in figure 6.1. Each parking space is equipped with one ultrasonic sensor and one LED. They are all connected through wires on

dedicated pins on an Arduino Uno which has a wired connection to a Raspberry Pi. The Arduino receives the measurement of each ultrasonic sensor and controls every LED light through integrated programming logic. The action of turning the light on if a car was parked is directly implemented in the physical layer and is not controlled by the application layer. The application layer is only informed about the changing parking status. So it can be said that, some control is given within the Arduino as well.

The physical layer of the smart car parking system includes several sensors and actuators, which communicate with each other and with the server. These devices are as follows.

A) Two license plate reader cameras: these cameras are responsible for reading the license plate number of a car and for sending the open / close commands to the automatic barrier. Data from the license plate reader of the entrance is used to indicate additionally the data of reserved location to the LED board.

B) Automatic barrier: this barrier receives the open / close commands and acts as an actuator that permits or prevents cars from entering the parking area. There is also another automatic barrier for the exit side of the parking.

C) The LED display: this LED screen acts as a guide for the driver. It receives information about available lots and displays it to the driver.

D) Ultrasonic Sensor: The ultrasonic sensors are connected to the Arduino with wires, the sensors detect whether the parking lot is occupied or vacant and display the LED lights. A red LED light indicates that the parking space is occupied / reserved and green LED light indicates that the parking space is vacant.

E) The Sink Node / Raspberry Pi: the sink node receives all the information from the parking lot sensor and passes this information to the main controller.

Network Layer:

The network layer establishes the connection between the physical and the application layer. In our smart car parking system, the data transfer is handled over a Raspberry Pi based Sink Node which uploads the data to a Cloud service. The Cloud service provides the data to the web and mobile application for further use. The Arduino gateway receives input sensor data and outputs actuation signals to the actuators. In detail, the physical data will be transferred from the Arduino Uno to a Raspberry Pi based Sink Node via a serial port through a USB cable. On the Raspberry Pi / Sink Node, a script will run which sends the received data from the Arduino as a series of continuous payloads to the cloud. For further extension of parking area, it is recommended to use Wi-Fi as wired connections are not feasible at a large parking area. Consequently, the Wi-Fi range of the router may not cover the whole parking area, therefore, the amplifier is mounted to boost and repeat the Wi-Fi signals so that all devices can access the internet.

Application Layer:

In the application layer, the smart parking system is used by end users via a mobile application and managed by an administrator via a web application. Data processing and integration over the entire set of individual parking areas is done through a controller in the application layer. The data is delivered to mobile application with the help of cloud. The mobile application is able to provide users with the current occupancy of parking spaces and lets them monitor their respective parking information. Further, the user is able to reserve a particular parking space beforehand. Both, user and admin, will log in to the same application. The admin's e-mail address will be saved in the database mapped to an admin role. Therefore, the system can distinguish between end user and admin. This is important, because they have different views based on functionalities and privileges.

6.1.1 Identifying Security Requirements of Smart Car Parking System

After analysing the system environment, we perform the CPS framework activities. Therefore, in the following section, we have applied the eight activities of the proposed framework and shown how they can be used to elicit security requirements effectively.

In order to fulfil the framework workflow process, the input, technique employed, and output of each activity should be determined. Once these eight activities are completed, the security requirements can be easily identified. We have applied our tool to determine security requirements for smart car parking system on the basis of our SRE framework. The framework has 8 activities and for every activity there is a predefined set of potential inputs, techniques and outputs. Being a specific domain / environment application of a very general framework, we have presented only the relevant and applicable inputs, techniques and outputs from the generic list provided in Chapter 5. At times, certain application specific elements have also been added. We have followed this procedure for each of the eight activities as outlined in Chapter 5.

A1: Identify assets

The first activity of the framework is to identify the system assets - all system related elements that hold significant value to the stakeholders. The workflow process for this activity, as described in Chapter 5 (section 5.3), is used as a guideline for implementing this activity. Our SRE tool provides a generic checklist of assets, which offers asset proposals from which we can identify the important / relevant / most valuable ones after analysing the system architecture of smart car parking system and discussion with primary stakeholders. Assets specific to the given system not listed in the tool may be added by the security analyst.

Input: The architecture of smart car parking system, checklist of assets

Technique: Facilitated meeting session (detail analysis, interview)

Output: After analysis of the car parking system entities, we have listed eleven assets for the smart car parking system.

S.Nr	List of Assets
1	Customer data
2	Ultrasonic Sensor
3	Camera
4	Controller/ Sink node
5	LED lights
6	LED display
7	Vehicular data
8	Server
9	User application
10	Barrier
11	Parking lot

Figure 6.3 shows how the SRE tool is used to select and finalize the list of assets for smart car parking system.

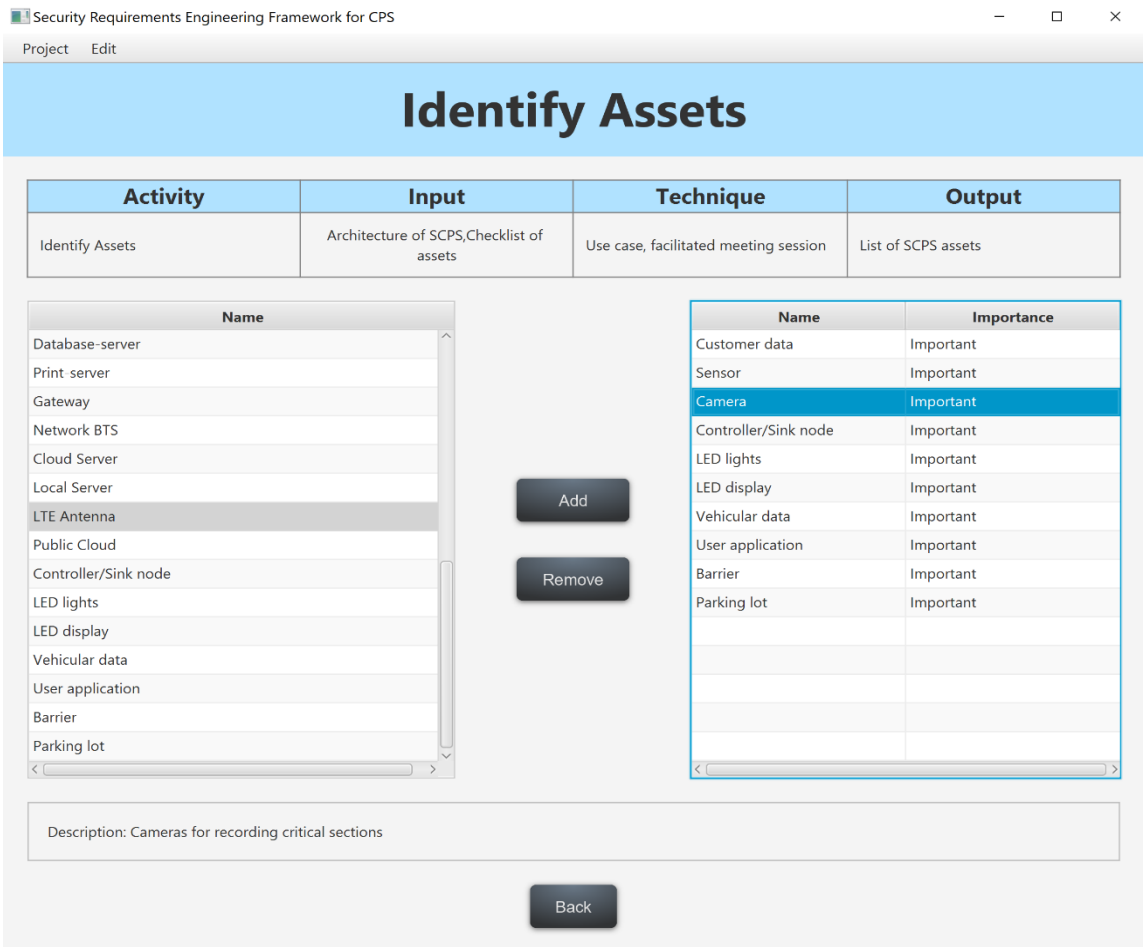

Figure 6.3 Asset Identification on SRE Tool

A2: Identify security goals

The purpose of this activity is to identify security goals for the smart car parking system. Our SRE tool provides a generic checklist of security goals, which can be shortlisted from based on an analysis of the assets derived from the previous activity and discussion with primary stakeholders.

Input: Checklist of security goals, list of assets from output of activity 1

Technique: Facilitated meeting session (detail analysis)

Output: List of identified security goals

S. Nr	List of Security Goals
1	Integrity
2	Availability
3	Confidentiality
4	Authorization

A3: Identify threats

This activity aims to identify the threats to the smart car parking system. The SRE tool offers an extensive list of potential threats as identified in the literature, some of which are discussed in Chapter 2. We apply the misuse case technique to these potential threats to analyse the relevance and impact of such threats, after which we are able to identify the ones most critical to the given smart car parking system.

Input: General checklist of CPS threats

Technique: Misuse case

Following are the major threats that we identified, using the tool to build misuse cases:

Threat analysis on Physical Layer: We analyse threats on the physical layer of the smart car parking system. The attacker may attempt to exploit / distort the functionality of the camera by causing damage to the camera. In this case, the camera is unable to read the vehicle registration number. The attacker may attempt to damage

the barrier, in this case the whole smart car parking system would collapse and no vehicles will be able enter/exit the parking area.

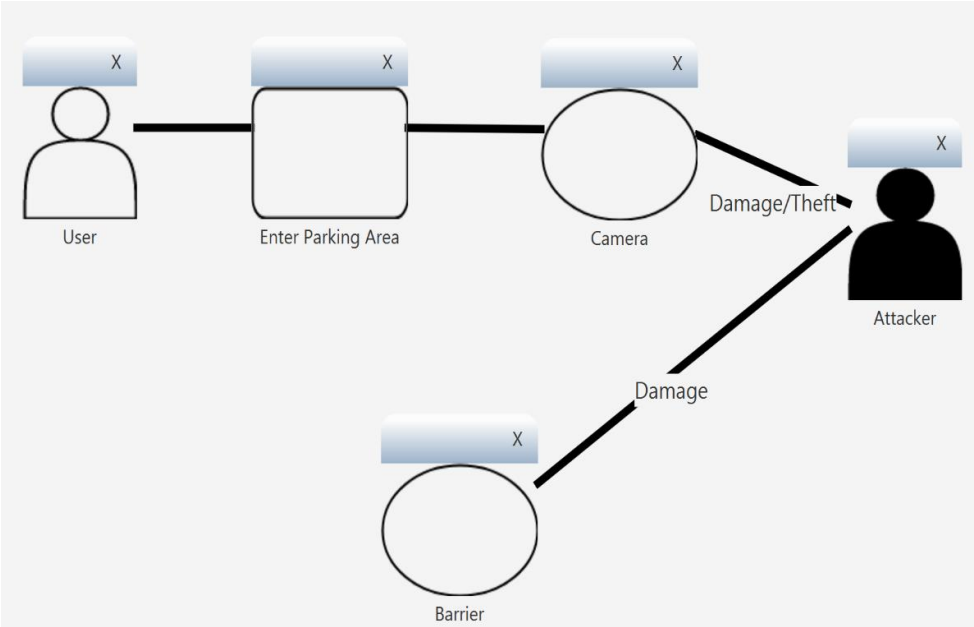

Figure 6.4 SCPS threats on physical layer 1 generated on SRE Tool

The attacker may attack the sensor by using jamming, malicious corrupted nodes, or using a battery draining attack as shown in figure 6.5. The attacker may also attack the sink node to divert the vehicle data to some unknown place. Furthermore, the attacker may attempt unauthorized access or damage on LED display and LED lights which affect the whole car parking system.

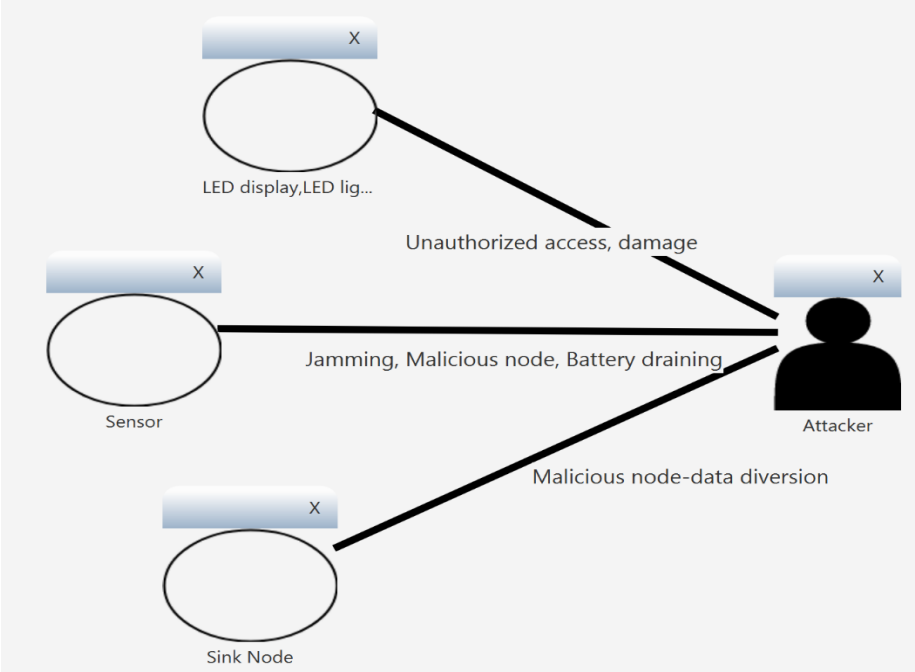

Figure 6.5 SCPS threats on physical layer 2 generated on SRE Tool

Threat analysis on Network Layer: We conducted threat analysis on the network layer, so that we can determine security requirements for smart car parking system with regards to the network layer. There is a real possibility that the attacker may attempt a DoS attack to stop or delay communication to and from the system website. The attacker can get the user or vehicle data by attempting the eavesdropping attack.

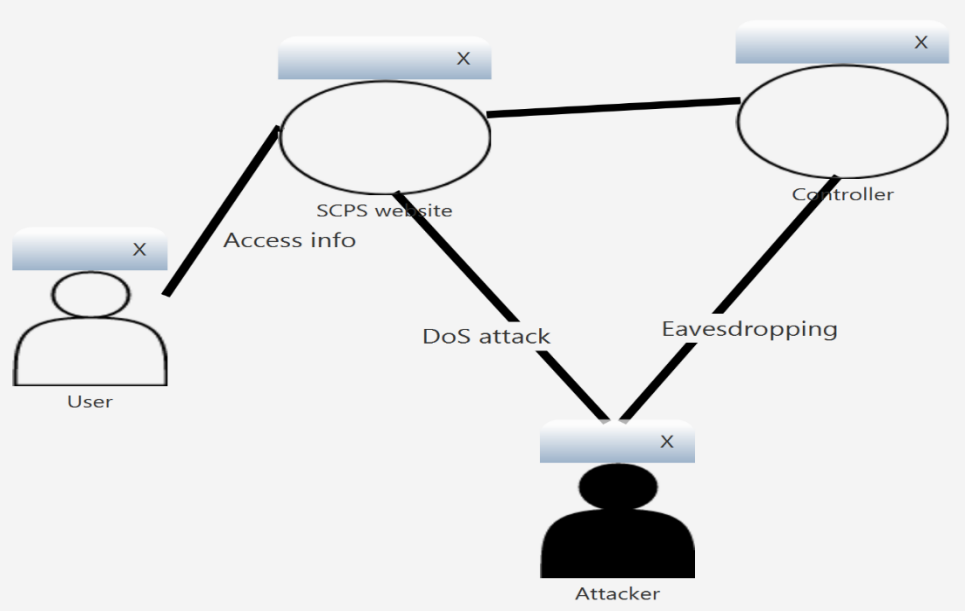

Figure 6.6 SCPS threats on network layer generated on SRE Tool

Threat analysis on Application Layer: The attacker may obtain confidential vehicle data by using unauthorized access or may insert malicious software to download user credentials information for blackmail / threatening purposes.

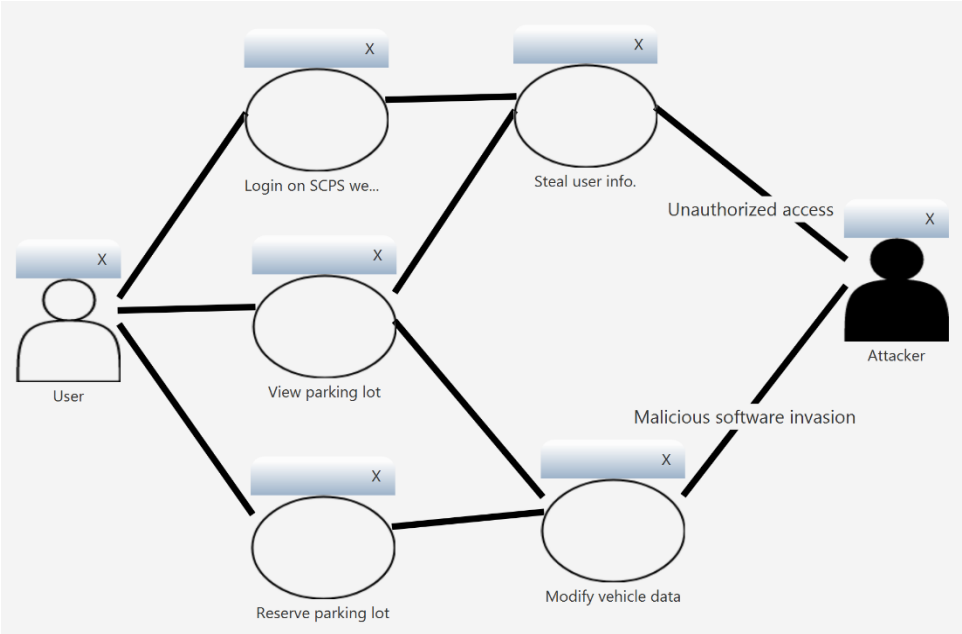

Figure 6.7 SCPS threats on application Layer generated on SRE Tool

After using the technique of misuse case, we are able to determine the major threats faced by a smart car parking system by shortlisting from a generic checklist. For example, the generic checklist provided in the tool lists damage as a prominent threat. Given that the barrier is an important asset, damage to the barrier becomes a threat to the system. This can be materialized in the form of an individual causing a heavy object to impact with the barrier for example, which would cause it to lose its functionality. The same argument could be made for the camera as well. Following this kind of line of thought, we are able to identify all threats.

Output: List of identified threats

S. Nr	List of CPS Threats
1	Corrupted/Malicious Node Attack
2	Malicious node-data diversion
3	Denial of Service (DoS)-Network
4	Eavesdropping
5	Unauthorized Access
6	Denial of Service (DoS)-Jamming
7	Damage/Theft
8	Vandalism
9	Battery draining
10	Failure/Malfunctioning
11	Malicious software
12	Packet modification

A4: Identify secure network communication

This activity involves identifying network communication channels that are both secure and feasible for our system. The SRE tool contains a checklist of communication protocols that can be used in a CPS, and through analysis of system architecture, system components in the three respective layers and the nature of the communication involved alongside the threats it may face, the most appropriate communication protocol(s) are selected. This includes an analysis of data sensitivity, logistical feasibility and hardware capabilities of the respective components.

Input: List of secure network communication

Technique: Facilitated meeting sessions (analysis and comparison)

Output: Identified list of Secure Network Communication

S.Nr	List of Secure Network Communication
1	Transport Layer Security (TLS)
2	Secure Sockets Layer (SSL)
3	Hypertext Transfer Protocol Secure (HTTPS)
4	Secure Electronic Transaction (SET)
5	Point-to-Point Protocol (PPP)

A5: Identify hardware endpoint

This activity determines main hardware endpoint. Based on the system architecture, we assess for suitable hardware endpoint from a general checklist provided in the SRE tool. The analysis is focused on careful selection that ensures full functionality as well as vendor repute and authenticity.

Input: Checklist of hardware endpoint

Technique: Facilitated meeting sessions (analysis, group discussion)

Output: Identified list of hardware endpoint

S.Nr	List of Hardware Endpoint
1	Sink node/Raspberry pi
2	Controller
3	Network wire
4	Server
5	Barrier
6	Fastening Bar
7	Cable
8	Router
9	Gateway/Arduino
10	LED display
11	LED lights
12	Electronic panel box
13	Amplifier
14	Control box

A6: Identify sensor types and communication medium

This activity identifies suitable sensor types for the system and their corresponding communication media. From a general checklist of sensors and communication media, the appropriate selection is made based primarily on characteristics of security and reliability.

Input: List of sensor type and communication medium

Technique: Facilitated meeting sessions (analysis and comparison)

Output: Identified sensor types and communication medium

S.Nr	List of Sensor Types & Communication Medium
1	Ultrasonic Sensor
2	Wifi
3	IEEE 802.3 (Ethernet)

A7: Perform risk assessment

Here, we perform a risk assessment for each of the major assets identified in activity 1. Misuse cases are used to obtain information about not only the nature, avenues and kinds of attack that may be expected on an asset, but also about the expected likelihood of an attack as well as potential impact of the losses incurred as a result of the identified threats for the asset. Risk for each of the assets is calculated by multiplying the likelihood and impact of the respective threats related to the asset. This step combines data obtained from the previous activities and uses them to rank the threats in terms of their risk, with the high-risk threats and their associated assets at the top, so that greater priority may be given to ensuring risk mitigation in their regard.

Input: List of assets and threats

Technique: CPS Risk Matrix and Misuse case

Output: Identified risk

S.Nr	List of risk based assets	Likelihood	Impact	Risk
1	Barrier	Very High	Very High	Very High
2	Camera	High	Very High	Very High
3	Customer data	High	Very High	Very High
4	Sensor	High	High	High
5	Sink node	Medium	Very High	High
6	User app/website	High	High	High
7	Server	High	High	High
8	LED display	Medium	High	Medium
9	LED lights	High	Medium	Medium
10	Controller	Medium	High	Medium

The above risk results are generated using the CPS Risk Matrix.

A8: Elicit security requirements

In order to elicit security requirements for a smart car parking system, the security goals, assets, and threats need to be analysed, together with the security-risks. All security goals, assets, threats, and risks of the system are subjected to detailed analysis by the stakeholders through misuse case. To do this and having completed the first 7 activities proposed in our SRE framework, we aggregate the outputs of each of these activities to elicit the security requirements.

The SRE Tool is designed to export the results from these 7 activities onto a single file which makes the process of security requirements elicitation significantly easier. This result format is as presented in figure 6.8 below.

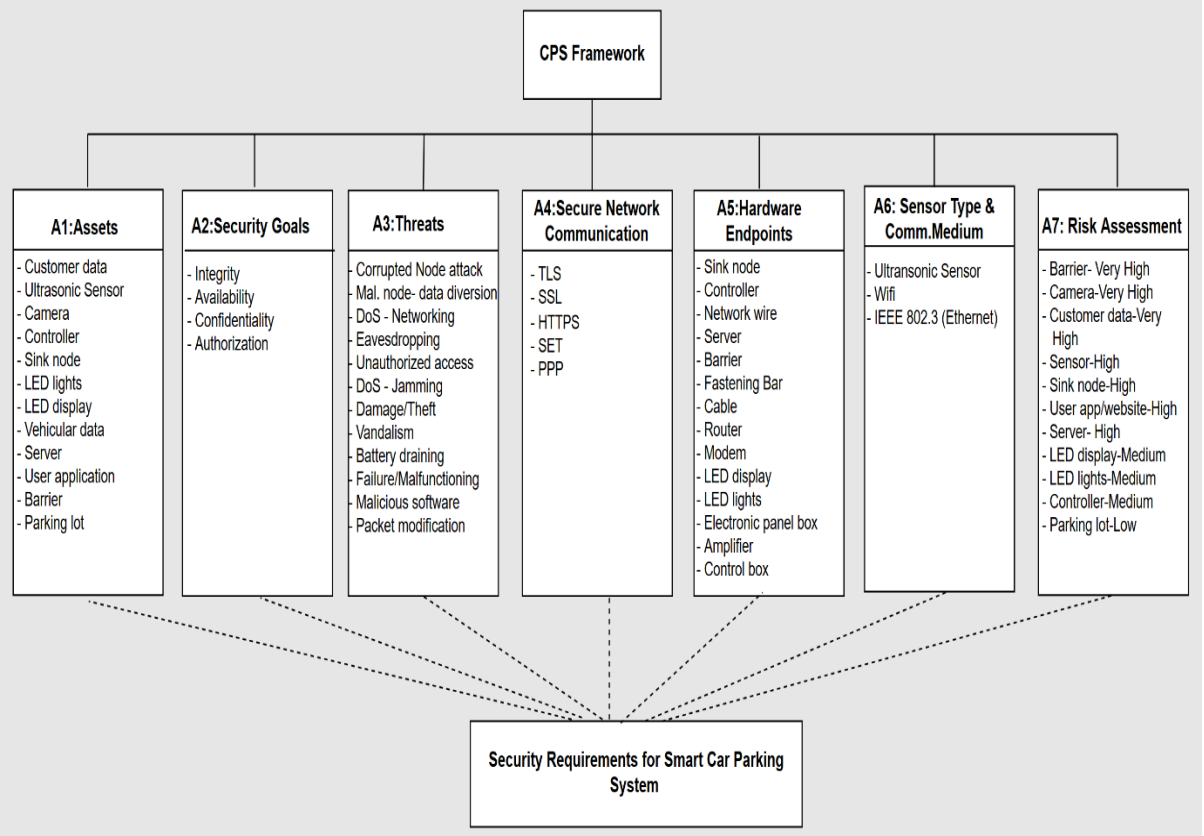

Figure 6.8 Output of all activities of smart car parking system

Keeping in view these results from the previous activities, we combine them together to form security requirements. This can be done by individually considering different assets, together with various aspects of security goals and threats, and discussing their security at length with the key stakeholders through the misuse case technique.

Let us consider an example from the physical layer of the CPS. One of our critical assets are the camera, for which the security goal of availability is of paramount importance. The possible threats to camera are damage, vandalism and malfunctioning, which we saw from the misuse case analysis, along with details of how and to what end such attacks could be enacted. A discussion with primary stakeholders revealed that damage or theft of the camera could lead to a breach of the availability security goal., which would compromise the system. To counter these threats, we determine the Security Requirements (SR) (numbers 1 and 2):

SR-1	In case of camera, barrier, sensor, or any other devices damage/theft, vandalism, malfunctioning, the system shall provide alarm notification, and informed to the administrator.
SR-2	There should be redundant camera and more than one kind of multiple power supply backups for the camera to prevent damage, vandalism, malfunctioning or any common mode failure.

This is shown through misuse case as shown in the figure 6.9:

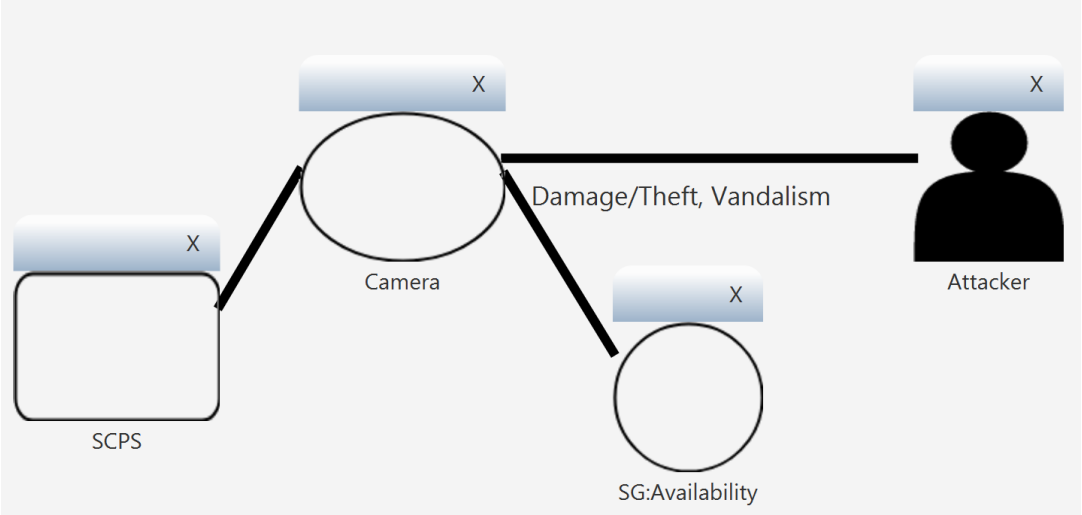

Figure 6.9 Eliciting security requirements of SCPS with misuse case (Physical Layer)

Taking a look at the network layer, one of the most important assets for the system is the server that ensures communication between the physical environment and sink node to the controller. Integrity and availability of this server are important security goals for the running of the system. The possible threats in this case are DoS attack and unauthorized access/modification, identified in misuse case. Discussion with stakeholders revealed that one way to do this is to overload the system with heavy files or large number of requests, which would cause lag or even cause the servers to go down completely, compromising the availability of the system. Unauthorized access could be done by repeated guessing for passwords by a malicious agent. To counter these threats, we determine the security requirements (number 18, 28):

SR 18	The system shall prevent unauthorized access/modification to the communication between a server and car parking controller.
SR 28	The system shall detect/prevent and mitigate the influence of DoS attacks on the server from outside the system, e.g. huge amount of user requests.

This is shown through misuse case in the figure below:

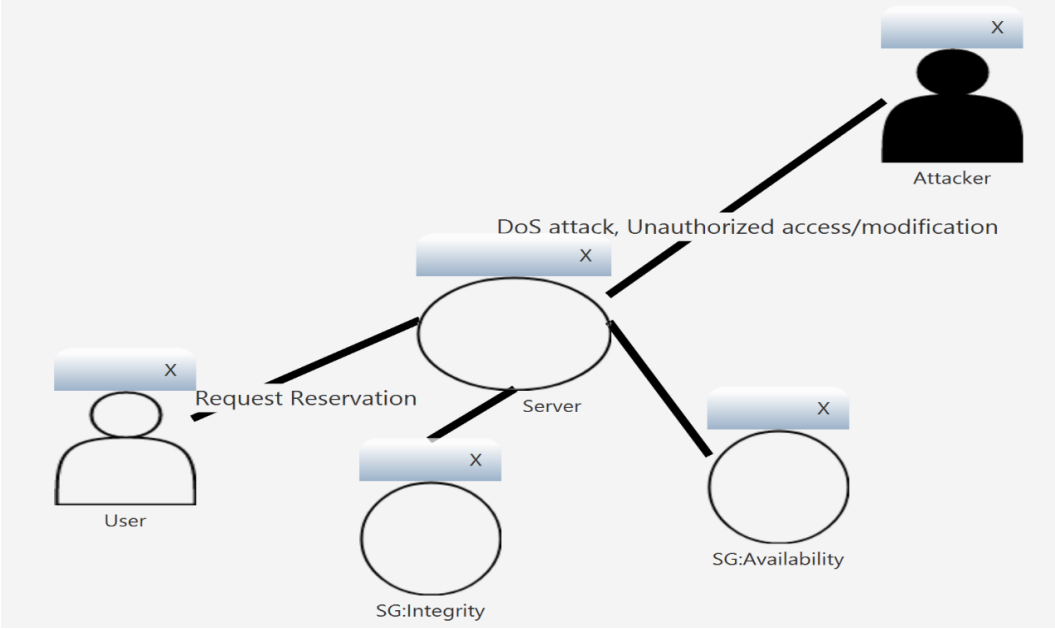

Figure 6.10 Eliciting security requirements of SCPS with misuse case (Network Layer)

Finally, let us consider one of our primary assets from the application layer, for example, customer data and user application. Among the security goals important to us is confidentiality. A threat that may seriously challenge this goal of confidentiality is malicious software and unauthorized access. The misuse case analyzed that malicious software may attempt to gain control of the user application, and through it, customer data by hiding as an advertisement or harmless downloadable file. To counter these threats, we determine the security requirement (Number 31, 34):

SR 31	The system shall prevent user data from unauthorized access throughout the system.
SR 34	The system shall prevent malicious software in the car parking website.

The misuse case is depicted in the figure 6.11.

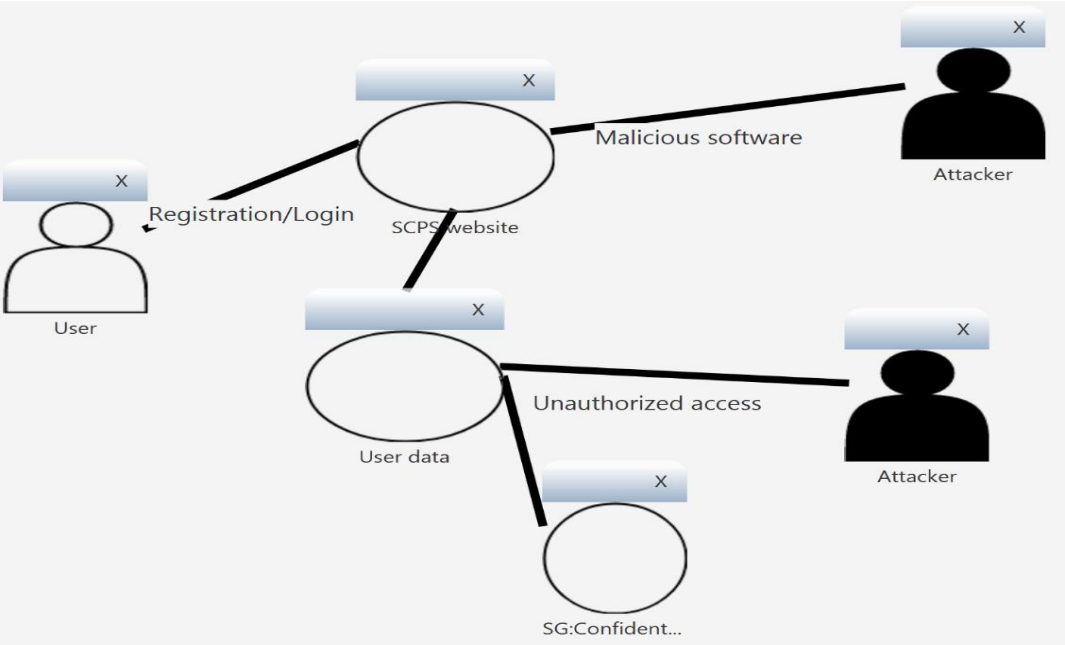

Figure 6.11 Eliciting security requirements of SCPS with misuse case (Application Layer)

Using the methodology outlined above, we have determined 43 security requirements for a smart car parking system, as shown in table 6. 1. Furthermore, we ignore those security requirements which are not feasible according to a cost-benefit analysis, i.e., those threats that have very low risk coupled with resource-intensive requirements for risk mitigation would then be excluded from the final list of security requirements. As a reasonable compromise, we consider only those security requirements which have very high to medium risk if there is a high cost associated with mitigating that risk.

For example, *"In case vehicle registration is stolen, a secure method must be implemented for vehicle authentication (e.g. camera technique involving image processing to scan registration number plate of vehicle and match with another entry in database like vehicle make or model)".* This security requirement aims to stop a relatively unlikely threat, which even if it occurs, would probably not compromise the entire system. Given this low risk, and the high cost of investing in alternate most elaborate authentication methods, it doesn't make sense to mandate this security requirement for the system.

Table 6.1 Security Requirements for Smart Car Parking System

ID	Security Requirements
	PHYSICAL LAYER
1	In case of camera, barrier, sensor, or any other devices damage/theft, vandalism, malfunctioning, the system shall provide alarm notification, and informed to the administrator.
2	There should be redundant camera and more than one kind of multiple power supply backups for the camera to prevent damage, vandalism, malfunctioning or any common mode failure.
3	The source of power, power-point connection and cables for camera shall not be reachable to general public and unauthorized person.
4	There should be an authorized manual opening provision of barrier in case of damage or failure of barrier.
5	The system shall prevent unauthorized access in the parking lot area, changes and updates in the camera, sensor, barrier, LED display, LED lights and actuator configurations shall exclusively be initialized and conducted by personnel with special security permissions.
6	The system shall prevent battery draining attempts on the sensor box.
7	In case of DoS (e.g. Jamming attack on sensor node) attack, the system shall detect/ prevent delays or asynchronous response times of sensors and actuators.
8	The system shall prevent/detect unauthorized access of sensors, LED display, LED lights in the parking lot area.
9	There shall be stand-by sensor for backup, in case for the sensor damage, theft, vandalism, malfunctioning or any common mode failure.
10	The system shall prevent malicious sensor node attack (in this case the sensor shall not be able to send the wrong parking lot information to the sink node).
11	The system shall monitor unauthorized replacement (e.g. malicious node) of sensors/actuators in the parking lot.
12	The sink node shall not divert the data into unknown server (e.g. malicious node attack-data diversion).
13	The sink node shall not be able to send wrong parking lot information to the server/controller.
14	All parking devices (i.e. camera, sensor, barrier, LED display, LED lights etc) shall be from well-known manufacturer.
15	There shall be physical protection and implementation of standards (e.g. IP, IEC, IEEE etc.) for installation of all hardware devices.
16	The system shall prevent unauthorized access, accident or damage of LED display, LED lights in the parking lot area.
	NETWORK LAYER
17	The camera shall use 802.3 standard wired Ethernet.
18	The system shall prevent unauthorized access/modification to the communication between a server and car parking controller.
19	The system shall use 'SSL/TLS' to secure authentication for server management.
20	The system shall use 'HTTPS' protocol to secure data communication on internet.
21	The system shall use 'SET' protocol to secure parking payment on internet.
22	The system shall detect/ prevent DoS attacks on the sensor networks.
23	The communication of parking barriers and parking sensors shall exclusively be limited to system components. Any communication attempts from outside the system shall be blocked.
24	A preferably wired medium shall be adopted in all localized areas control system (i.e. parking entry and parking lot area) to avoid hacking/distortion threats.
25	The car park system shall use point-to-point protocol between controller and user application.
26	The system shall prevent modification or deletion (e.g. packet modification) of transmitted data in the network.

27	The system shall prevent sniffing (e.g. by eavesdropping) and monitoring traffic on the communication links of sensor networks (i.e. camera, sensor and sensor node) and controller.
28	The system shall detect/prevent and mitigate the influence of DoS attacks on the server from outside the system, e.g. huge amount of user requests.
29	The system shall detect high traffic loads on parts of the sensor networks, server and initiate countermeasures.
30	The system shall deploy load relieving mechanisms when high communication traffic occurs on the parking system website.
	APPLICATION LAYER
31	The system shall prevent user data from unauthorized access throughout the system.
32	The system shall ensure that the user can securely access the parking system website and that sensitive user data are protected while using the website.
33	The system shall not provide user personal vehicle data information to any unauthorized person or organization.
34	The system shall prevent malicious software in the car parking website.
35	The system shall protect parking space status information from unauthorized access (e.g. malicious software) throughout the system.
36	The system database should have different access levels per stakeholders. Critical data should be encrypted and strong password protected.
37	The system shall be tested against malicious code, malware or other malicious software.
38	When using a newly installed app for the first time, the app shall request a login with user credentials. The user credentials shall then be authenticated by the system.
39	The system should ensure that the data which are required for requesting access to a reserved parking space shall only be downloaded by the app when the user is authenticated.
40	The system shall ensure that the user and vehicle billing data exclusively be managed by the admin.
41	The admin shall employ sanity and validity checks for processing user and billing data.
42	The system shall ensure that the confidential user data, i.e. passwords or payment information, and detailed billing information will not be revealed to any entity except of the respective user.
43	The system shall ensure that the security mechanisms and configurations shall exclusively be managed by the admin.

6.2 Case Study 2: Soccerwatch

The second case study is based on a real world scenario called Soccerwatch. Soccerwatch GmbH is a start-up firm based in Essen and founded in 2016 which sells smart camera systems to football clubs. In Germany, they already installed 75 of their camera systems. Some aspects of this concept are highly confidential due to security issues. Therefore, the information about Soccerwatch concept is provided on a high-level basis and neglects details such as the technical details of exchanged data and used components.

In general, the Soccerwatch camera system records the video and audio of a match and sends the data to the Soccerwatch team, which evaluates the material and provides it to the viewers. A smart camera system includes several cameras, audio, and temperature sensors and uses embedded Artificial Intelligence (AI) to follow the game and listen to the referee's commands. The AI identifies key points / highlights in the match and allows viewers to jump to points of interest, such as goals or penalties through the ninety-minute match. The camera system is placed at the middle of a football court connected to the power supply of the yard as shown in the figure. It sends video via 4G mobile data to the local server to cloud where the data will be processed and distributed to the internet platform through public cloud which the users can access via computer or mobile phone. Soccerwatch provides a live stream of football matches as shown in the figure 6.12. The highlights are identified through the evaluation of audio signals in the recordings.

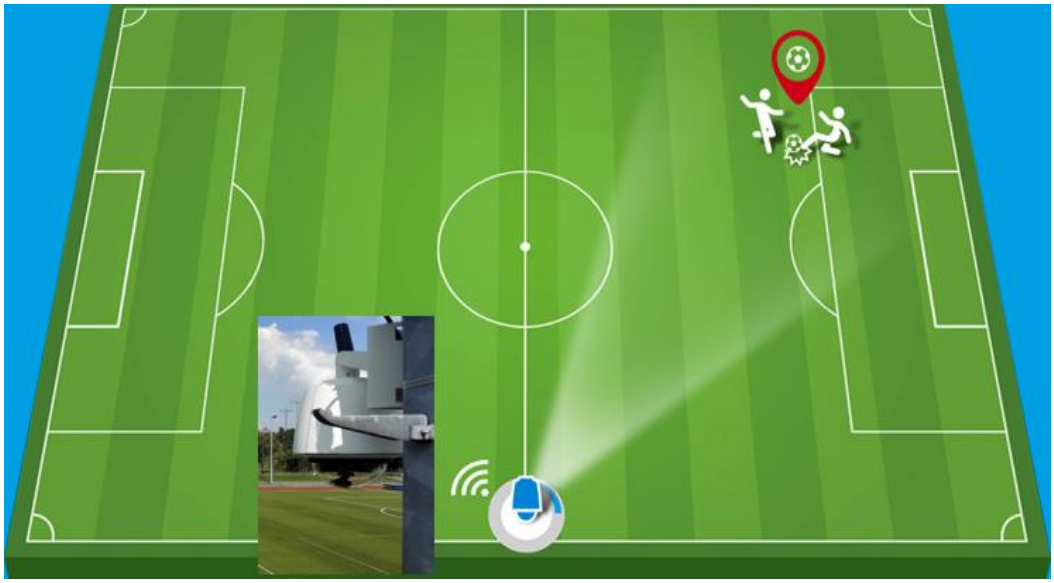

Figure 6.12 Soccerwatch live stream camera [226]

A trainer tool to track the players and analyze the game will be published in the future which also interacts with the CPS infrastructure. User can sign up to the system to access the trainer tool, a service which is handled separately from the live stream processing and the main CPS components. The sensed data are video, audio, and temperature. These three measurements influence the traffic in the CPS. The system chooses the video frame fluidly from one of the eight cameras at any given time corresponding to the camera that best shows the region of interest in the football field. The audio material influences the highlighting of the match events and the temperature sensor serves to trigger an alarm or shutdown the Soccerwatch camera to prevent malfunction under extreme temperature conditions. After understanding the Soccerwatch environment by creating few use cases, we design the Soccerwatch architecture as shown in figure 6.13.

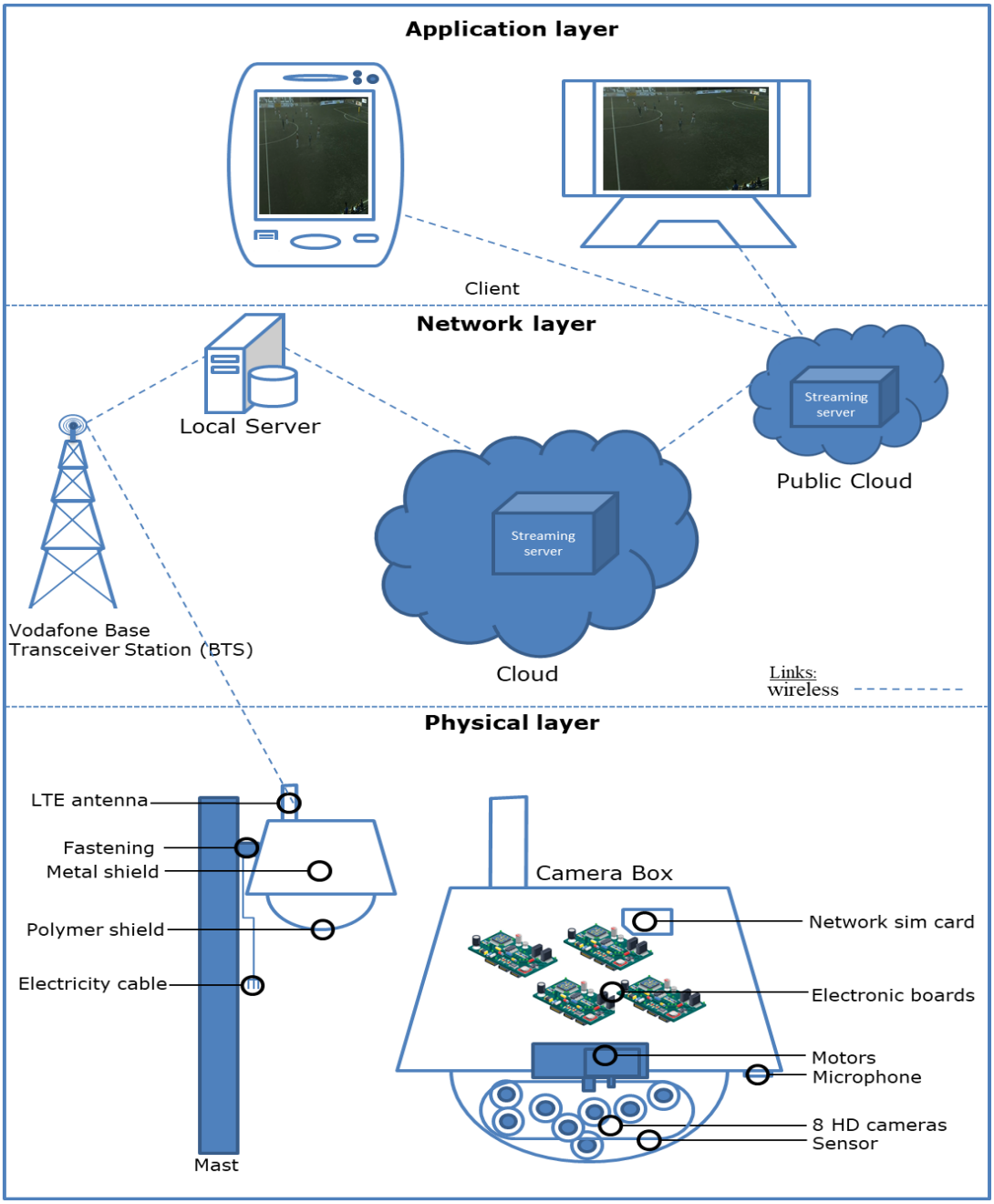

Figure 6.13 Architecture of Soccerwatch

Physical Layer:

Soccerwatch provides a comprehensive physical system package of different sensors to capture the soccer games which is further referred as the Soccerwatch camera. The Soccerwatch camera is usually attached to a mast at a football field, where it is connected through a cable to a power supply. It also has a cable connection to the electricity box in case of issues, where the system needs to be maintained or fixed on-

site. The Soccerwatch camera box includes multiple camera sensors, an audio sensor (microphone) for audio recordings, a temperature sensor connected on a processor board with an internet access interface. If a specific temperature has been measured that is too high for the camera to operate, the camera will be turned off to protect itself from overheating and physical damage. The cameras are recording the match in video format and microphones are recording the sounds. The cameras adjust the video recording to the light irradiation in the physical environment.

Network Layer:

Software updates and the stream is all managed over the 4G connection. The data from the camera sends video via 4G mobile to the local server to cloud where the data will be processed and distributed to the internet platform through public cloud which the users can access via computer or mobile phone. The transmission is conducted via 4G mobile data over a SIM card. The signal will be sent over the attached antenna to the next mast of the internet provider. The regular software updates will be provided over the mobile data.

Application Layer:

In the application layer is the control instance of the CPS in the form of a dashboard that displays all the Soccerwatch cameras and administrates their settings, e.g., scheduling of their actions. The time to turn on / off the camera is controlled by looking up the teams' websites for game schedules or on demand requests. Whenever needed, a camera starts recording a football match and provides the video to the viewers. In the cloud, Artificial Intelligence (AI) is applied according to the information it receives from the physical layer. The control instance, which is managed by the AI actuates the screen movement (zooming and screen detection). This will be sent as commands back to the physical layer. The network of the computer systems is currently controlled on-site. In order to improve the accurateness and to get low False-Positive Rate (FPR) from the measured data, the AI is continuously being trained with new data sets. In

the future, the measured sound data, and video data will be matched for a better result. Currently, the viewers can access the (live stream) videos via the internet website. An Android application is coming soon. User authentication is needed for registering for the trainer tools.

6.2.1. Identifying Security Requirements for Soccerwatch

After analysing the Soccerwatch environment, we perform the CPS framework activities. Therefore, in the following section, we have applied the 8 activities of the proposed framework and shown how they can be used to elicit security requirements effectively.

In order to fulfil the framework workflow process, the input, technique employed, and output of each activity should be determined. Once these eight activities are completed, the security requirements can be identified. We have applied our tool to determine security requirements for Soccerwatch. It must be noted that due to our agreement of confidentiality with our partner, certain sensitive details concerning the details of our study with Soccerwatch have been omitted from this thesis. The tool has 8 activities and for every activity there is a predefined set of potential inputs, techniques and outputs. Being a specific domain / environment application of a very general framework, we have presented only the relevant and applicable inputs, techniques and outputs from the generic checklist provided in Chapter 5. At times, certain application specific elements have also been added. We have followed this procedure for each of the eight activities as outlined in Chapter 5.

A1: Identify assets

The first activity of the framework is to identify the system assets - all system related elements that hold significant value to the stakeholders. The workflow process for this activity, as described in Chapter 5 is used as a guideline for implementing this activity. Our SRE tool provides a generic checklist of assets, which offers asset proposals from which we can identify the important / relevant / most valuable ones after analysing

the system architecture and discussion with primary stakeholders This included having in-depth discussions with representatives of the Soccerwatch company and concluding the results accordingly.

Input: The architecture of Soccerwatch, checklist of assets

Technique: Facilitated meeting sessions (detail analysis, interview)

Output: List of identified assets

S.Nr	List of Soccerwatch Assets
1	HD Camera
2	Vodafone BTS
3	Camera Box
4	Server
5	LTE Antenna
6	Mast
7	Microphone
8	Temperature Sensor
9	Soccerwatch data
10	Club member
11	User application
12	Admin

A2: Identify security goals

The purpose of this activity is to identify security goals for Soccerwatch. Our SRE tool provides a generic checklist of security goals, which can be shortlisted from based on an analysis of the important assets derived from the previous activity and discussion with primary stakeholders. Following a lengthy interview with the Soccerwatch representatives, each of the identified assets is matched with each of the generic security goals and those that are important are listed.

Input: Checklist of security goals, list of assets from output of activity 1

Technique: Facilitated meeting session (detail analysis)

Output: List of identified security goals

S. Nr	List of Security Goals
1	Integrity
2	Availability
3	Confidentiality

A3: Identify Threats

This activity aims to identify the most important threats to the system. The SRE tool offers an extensive list of potential threats as identified in the literature, some of which are discussed in Chapter 2. We apply the misuse case technique to these potential threats to analyse the relevance and impact of such threats, after which we are able to identify the ones most critical to the given Soccerwatch.

Input: General checklist of CPS threats.

Technique: Misuse case

Threat Analysis on Physical Layer: In the physical layer, we analysed the camera box, camera, microphone and sensor which were most vulnerable to the possibility of attacks from an external attacker. Put simply, we select the threat from the threat checklist and consider it in the context of these devices. This helps us to identify the threat. The attacker may try to damage the camera, try to stop the live streaming using DoS (laser light) attack, establish unauthorized access on microphone or exploit the functionality of the sensor using hardware Trojans attack as shown in the figure below.

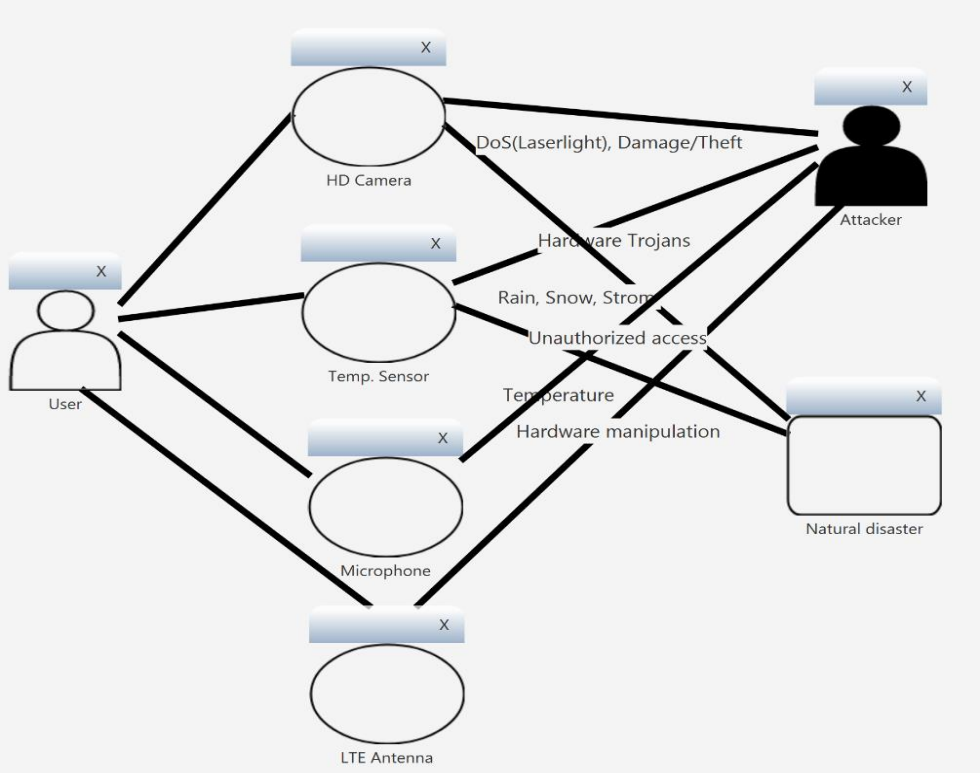

Figure 6.14 Soccerwatch threats on physical layer generated on SRE Tool

Threat Analysis on Network Layer: Figure 6.15 shows that the attacker may pose a threat to the local server using SQL injection which affects the performance of live video matches or Man-in-the-Middle attack to expose the confidential information of Soccerwatch, it may also try establish DoS attack on Vodafone BTS, in order to stop or delay the live streaming of Soccerwatch.

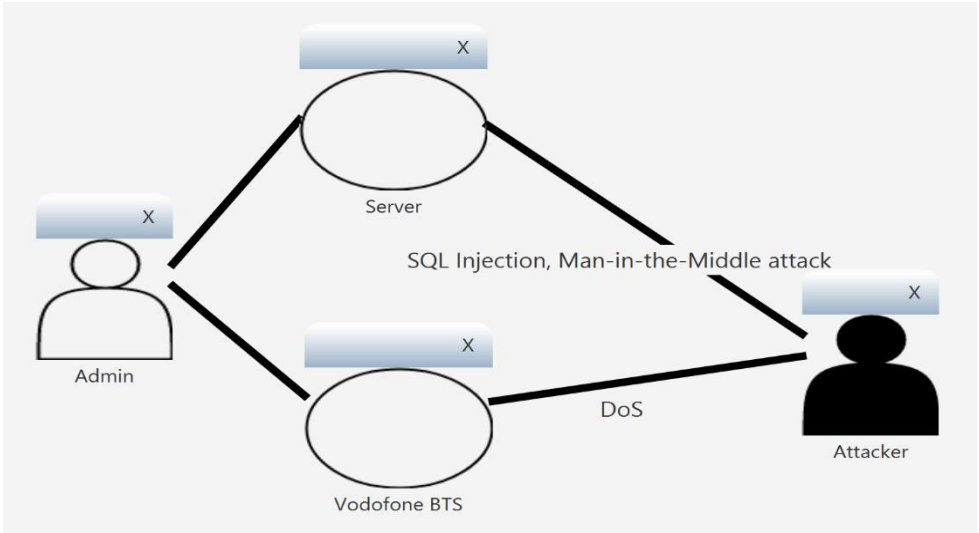

Figure 6.15 Soccerwatch threats on network layer generated on SRE Tool

Threat Analysis on Application Layer: In application layer, the attacker may try to manipulate the club member data for their own interest. The attacker will attempt to disclose the sensitive information of Soccerwatch for bad organization reputation in the market.

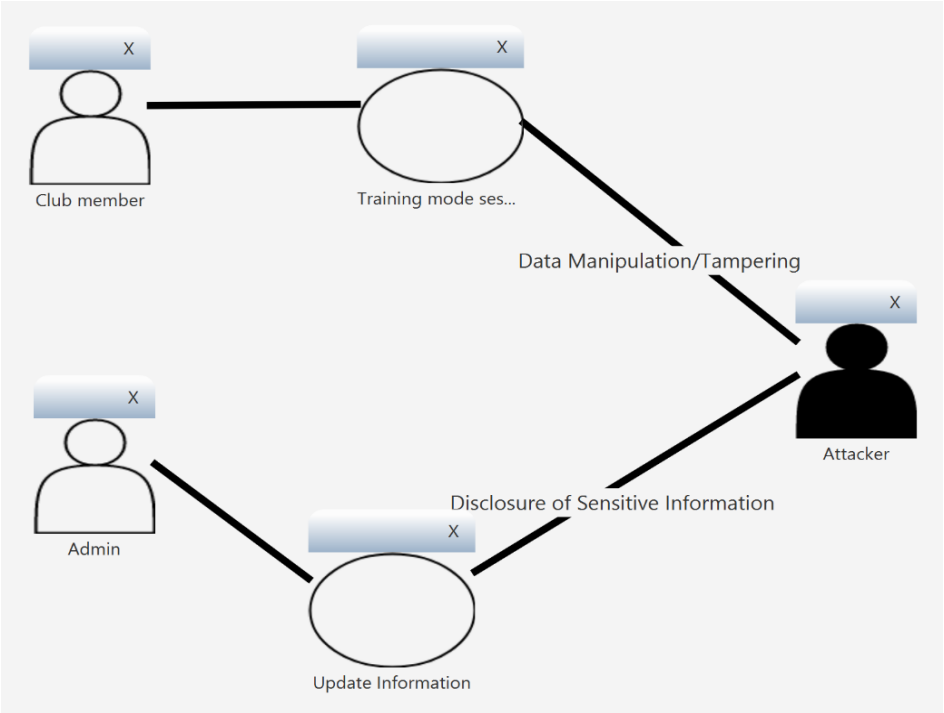

Figure 6.16 Soccerwatch threats on application layer generated on SRE Tool

After using the technique of misuse case, we are able to determine the major threats of Soccerwatch.

Output: List of identified threats of Soccerwatch

S. Nr	List of CPS Threats
1	DoS – Laser Light
2	Damage/Theft
3	Failure/Malfunction
4	Hardware Trojans
5	Unauthorized access
6	Natural disaster
7	Hardware manipulation
8	SQL injection
9	Man-in-the-Middle attack
10	DoS –Network
11	Data Manipulation/Tampering
12	Disclosure of sensitive information

Figure 6.17 shows how the SRE tool is used to identify and list the applicable security threats for the Soccerwatch.

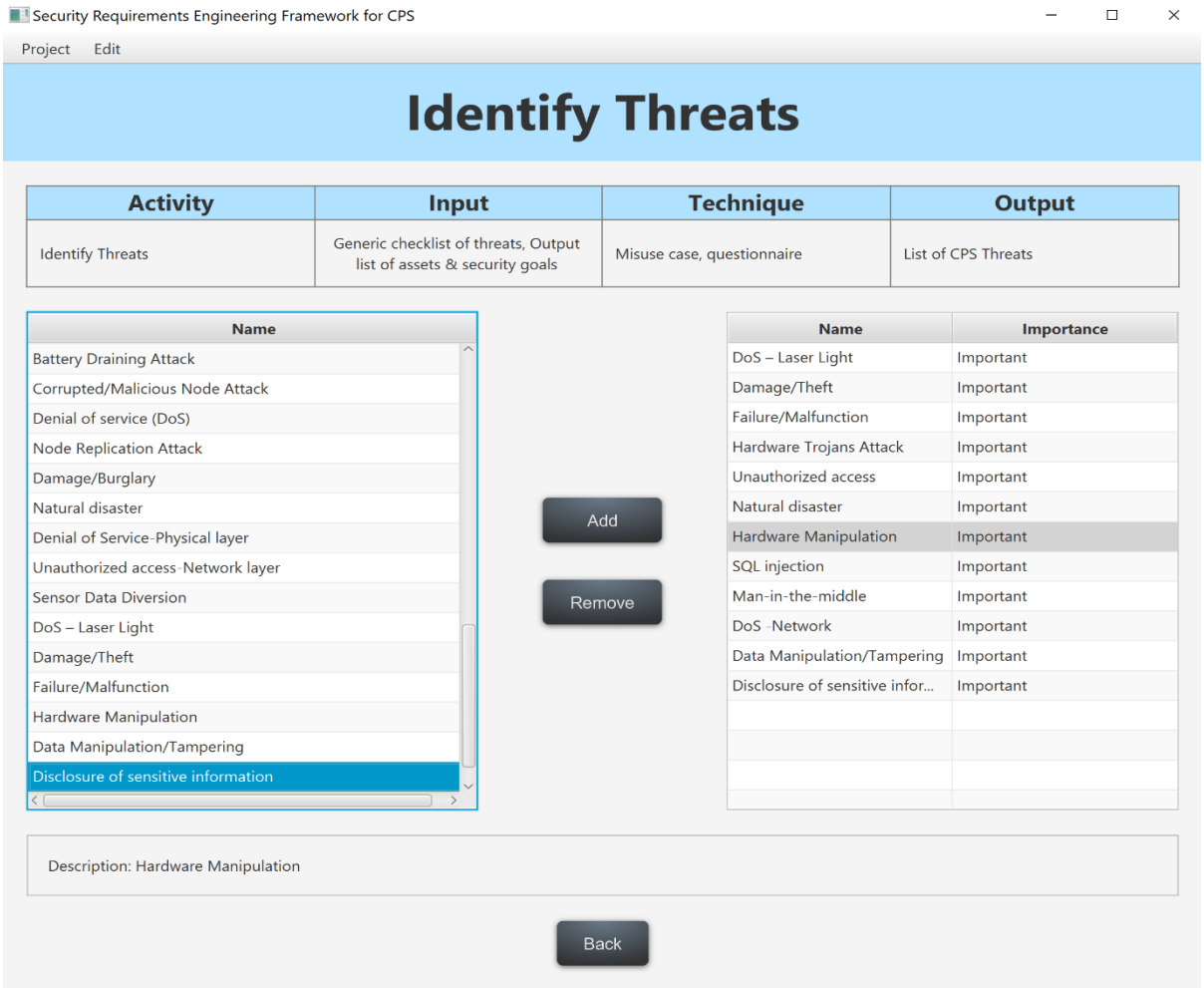

Figure 6.17 Threat Identification on SRE Tool

A4: Identify Secure Network Communication

This activity involves identifying network communication channels that are both secure and feasible for our system. The SRE tool contains a list of communication protocols that can be used in a CPS, and through analysis of system architecture and the threats it may face, the most appropriate communication protocol(s) are selected. The architecture of the Soccerwatch system is analysed and discussed with the company representatives. Threats from the previous activity are utilized to be aware of possible mechanisms of attack, and the most suitable and secure communication protocol is selected from the given list of network communication protocols in the SRE Tool.

Input: List of secure network communication

Technique: Facilitated meeting sessions (analysis and comparison)

Output: List of identified secure network communication

S. Nr	List of Secure Network Communication
1	Real-time Transfer Protocol (RTP)
2	Real-time Streaming Protocol (RTSP)
3	Hypertext Transfer Protocol Secure (HTTPS)

A5: Identify hardware endpoint

This activity determines main hardware endpoint. Based on the system architecture, we assess for suitable hardware endpoint from a general checklist provided in the SRE tool. This activity focuses on careful selection that ensures full functionality as well as vendor repute and authenticity.

Input: Checklist of hardware endpoint

Technique: Facilitated meeting sessions (analysis, group discussion)

Output: List of identified hardware endpoint

S. Nr	List of Hardware Endpoint
1	Microphone
2	Mast
3	Polymer Shield
4	Electricity cable
5	Fastening
6	Metal Shield
7	LTE antenna
8	Electronic Board
9	Network wire
10	Motor
11	Micro-controller
12	Cable
13	Gateway
14	Source power
15	Camera case

A6: Identify Sensor Types and Communication Medium

This activity identifies suitable sensor types for the system and their corresponding communication media. From a general list of sensors and communication media, the appropriate selection is made based primarily on characteristics of security and reliability.

Input: Generic types of sensor, checklist of sensor communication medium

Technique: Facilitated meeting sessions (analysis and comparison)

Output: List of identified sensor type and communication medium

S. Nr	List of Sensor Types & Communication Medium
1	HD Camera
2	Temperature Sensor
3	LTE 4G

A7: Perform Risk Assessment

Here, we perform a risk assessment for each of the major assets identified in activity 1 of Soccerwatch. Misuse cases are used to obtain information about not only the nature, avenues and kinds of attack that may be expected on an asset, but also about the expected likelihood of an attack as well as potential impact of the losses incurred as a result of the identified threats for the asset. Risk for each of the assets is calculated by multiplying the likelihood and impact of the respective threats related to the asset of Soccerwatch. This step combines data obtained from the previous activities of Soccerwatch and uses them to rank the threats in terms of their risk, with the high-risk threats and their associated assets at the top, so that greater priority may be given to ensuring risk mitigation in their regard.

Input: List of assets and threats

Technique: CPS Risk Matrix and Misuse case

Output: List of identified risk

S. Nr	List of risk based assets	Likelihood	Impact	Risk
1	HD Camera	Very High	Very High	Very High
2	Camera Box (Sensor, Microphone)	Very High	Very High	Very High
3	Mast	Medium	Very High	High
4	Server	Medium	Very High	High
5	Soccerwatch data	High	High	High
6	Club member	Medium	High	Medium
7	Admin	Medium	High	Medium
8	User app	Medium	High	Medium
9	Vodafone BTS	Low	High	Low

The above risk results of Soccerwatch are generated using the CPS Risk Matrix.

A8: Elicit Security Requirements

In order to elicit security requirements for Soccerwatch, the security goals, assets, and threats need to be analysed, together with the security-risks. All security goals, assets, threats, and risks of the system are subjected to detailed analysis by the stakeholders through misuse case. To do this and having completed the first 7 activities proposed

in our SRE framework, we aggregate the outputs of each of these activities to elicit the security requirements.

The SRE Tool is designed to export the results from these 7 activities onto a single file which makes the process of security requirements elicitation significantly easier. This result format is as presented in figure 6.18 below.

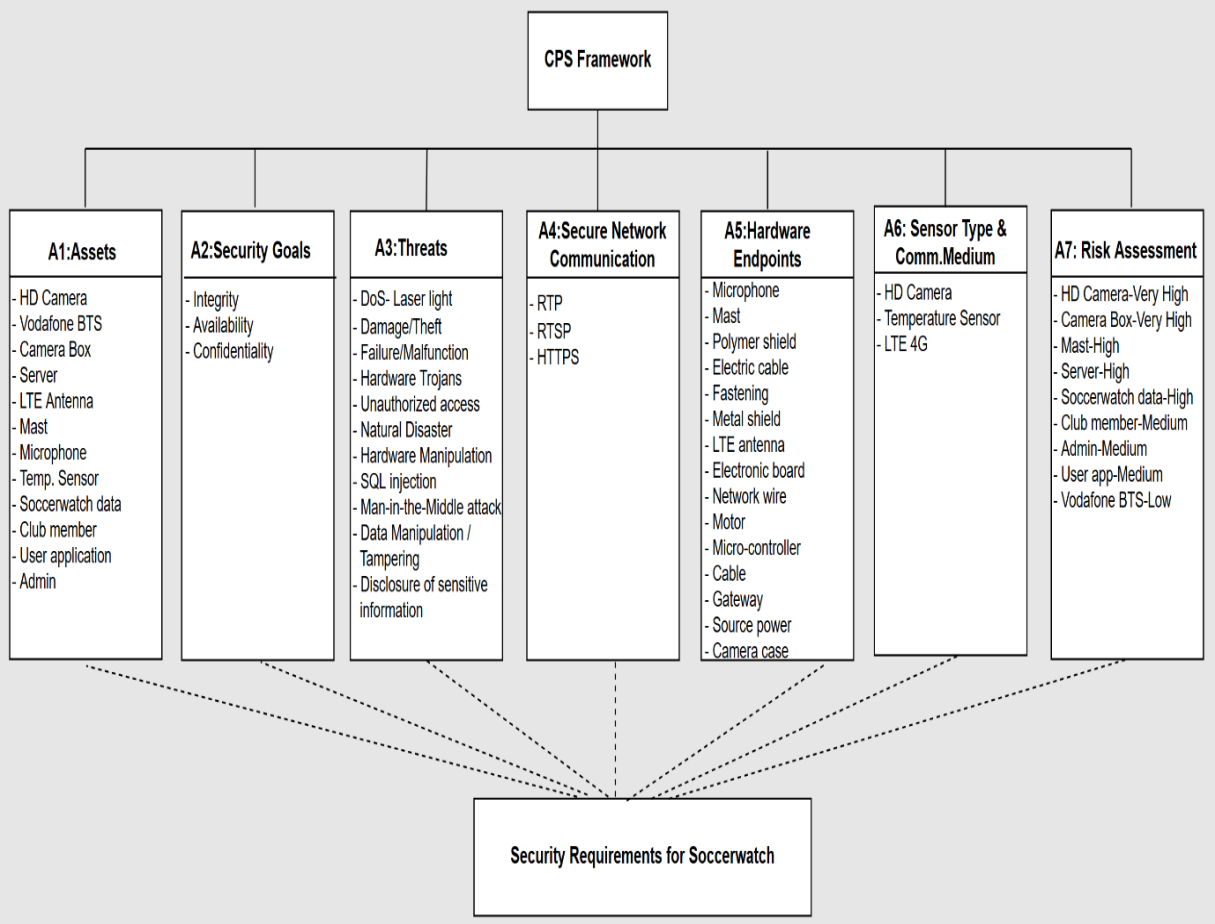

Figure 6.18 Output of all activities for Soccerwatch

In order to finalize security requirements for the Soccerwatch, we have utilized a very similar procedure to the one detailed in the smart car parking system case study. Below are some examples to illustrate the procedure in the context of this case study.

The most important asset from the CPS physical layer is the HD camera used for live-streaming, availability of which is an important security goal. Threats in this context

can be DoS (laser light) or damage/theft to the camera itself. To counter these threats, we determined the security requirements (numbers 1, 2, 6 and 7).

SR 1	The system shall ensure that the HD camera shall be protected against physical damages.
SR 2	The system shall ensure that the HD camera should be protected against theft. The assembly must be kept that its case can only be accessible to open through specific keys, also no screw or nut shall be provided to be open easily.
SR 6	The system shall prevent DoS attack (e.g. laser light) or any remote malicious attempt on camera.
SR 7	The system shall ensure that the HD camera should not be sensible against extra lights like laser.

This misuse case is depicted in the figure below:

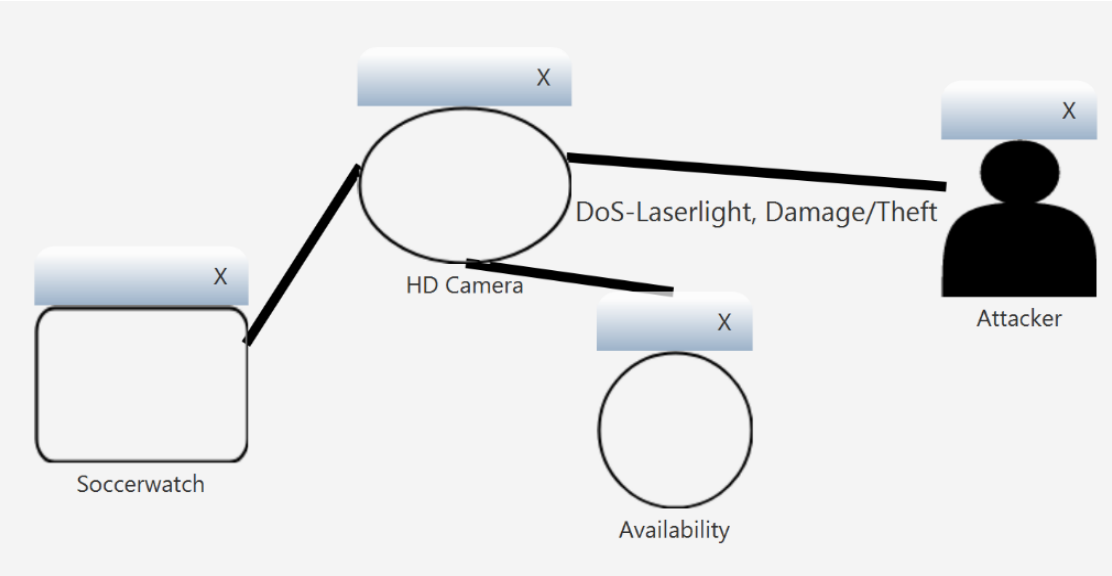

Figure 6.19 Eliciting security requirements of Soccerwatch with misuse case (Physical Layer)

Let us consider another example, this time from the network layer. The server is the important asset of Soccerwatch and it should be protected and integrity is crucial security goal. Threats to these goals can be in the form of SQL injection or Man-in-the-Middle attacks to get or compromise the Soccerwatch data. To counter these threats, we determined the security requirement (number 29).

SR 29	The system shall prevent any malicious attempts (SQL injection or Man-in-the-Middle attack) on server, in this case the system shall notify to admin.

This misuse is depicted in the figure below.

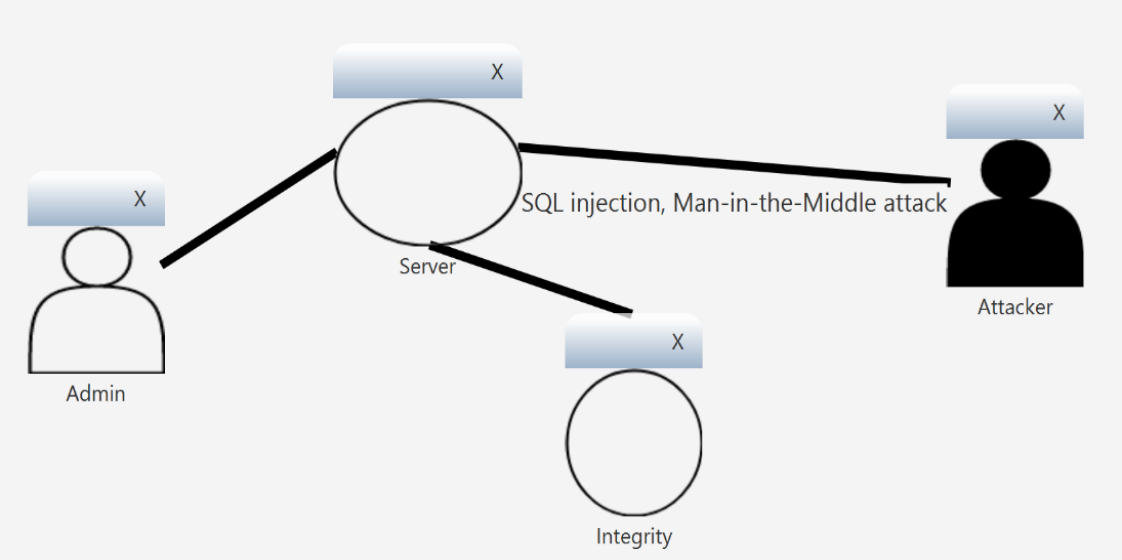

Figure 6.20 Eliciting security requirements of Soccerwatch with misuse case (Network Layer)

Let us look at an example from the application layer. One of the critical assets of the Soccerwatch is the club member, which holds the data for processing. Integrity of this data against modification is an important security goal for the system to function correctly. Unauthorized access can prove to be a major threat to the integrity of data held on the Soccerwatch. To counter this threat, we determine the security requirement (number 36):

SR 36	The system shall not disclose club member's sensitive information to unauthorized person.

This misuse case is depicted in the figure below.

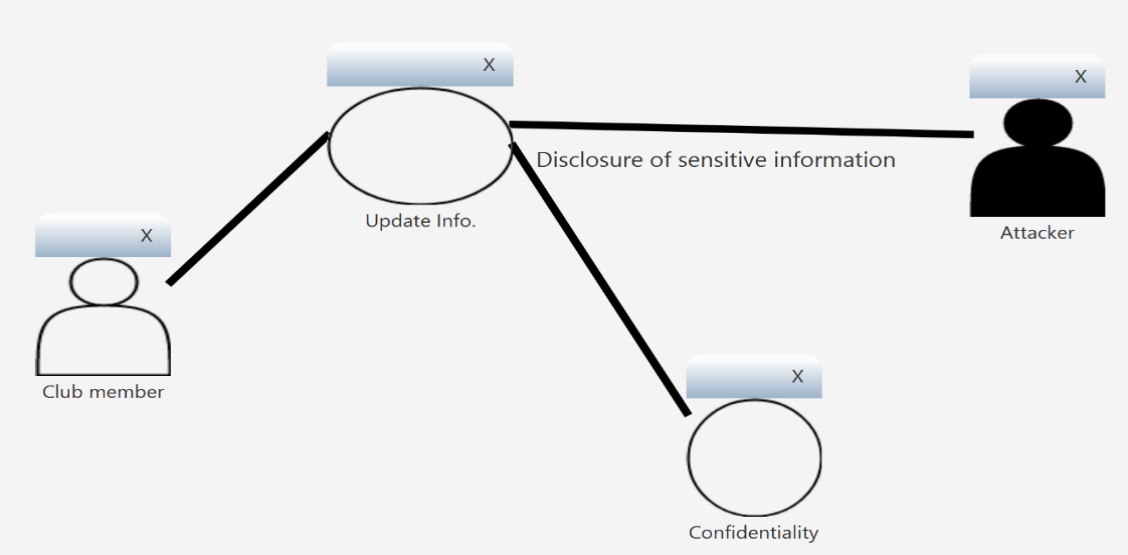

Figure 6.21 Eliciting security requirements of Soccerwatch with misuse case (Application Layer)

Using the methodology outlined above, we have determined 42 security requirements for Soccerwatch as shown in table 6.2.

Table 6.2 Security Requirements for Soccerwatch

ID	Security Requirements
	PHYSICAL LAYER
1	The system shall ensure that the HD camera shall be protected against physical damages.
2	The system shall ensure that the HD camera should be protected against theft. The assembly must be kept that its case can only be accessible to open through specific keys, also no screw or nut shall be provided to be open easily.
3	The system shall ensure that the height of HD camera should be place higher and cannot be easily reachable to general public or unauthorized people.
4	The system shall ensure that the camera box, source power, termination box, connections and any other associated devices should not be open access or reachable to general public or unauthorized people.
5	The system shall ensure that there should be more than one kind of multiple power supply backups for the camera to avoid failure or malfunctioning.
6	The system shall prevent DoS attack (e.g. laser light) or any remote malicious attempt on camera.
7	The system shall ensure that the HD camera should not be sensible against extra lights like laser.
8	The system shall ensure that in case of malfunctioning of any one of individual camera, the system shall notify to admin.
9	The system shall ensure that in case of stopping of any camera due to failure/malfunctioning or threat act the stand-by or redundant camera shall remain working.
10	The system shall prevent hardware Trojans attack to camera box to avoid any exploitation.
11	The system shall ensure that the camera box and mast shall be well protected against natural disaster e.g. very high or very low temperatures, windy, heavy snow, rain, dust or even earth quick tolerant.

12	A sensor shall observe the temperature of camera box, so that in extreme weather condition, this sensor powered off the camera and notify to admin.
13	The system shall monitor, record and report an unauthorized activity on camera box.
14	The system shall ensure that the mast shall be well protected against physical damage.
15	The system shall ensure that the LTE antenna should be protected against natural disaster.
16	The system shall ensure that the LTE antenna should be protected against hardware manipulation or physical damages.
17	The system shall ensure that the microphone should place on a protected area.
18	The system shall monitor, record and report an unauthorized activity on camera box.
19	The system shall ensure that the hardware upgrades/replacement should not have any negative effects on connectivity.
20	The system shall ensure that all Soccerwatch hardware parts shall be from well-known manufacturer.
21	The system shall ensure that the whole camera system including cameras transmission medium and other associated devices shall not provide easy or common interoperability medium for general public or unauthorized person. However, the standards should be maintained to attain technical interoperability.
22	The system shall ensure that the all hardware devices of Soccerwatch shall be configured/proper installation before placing it. Hence, minimize the vulnerability into the system.
	NETWORK LAYER
23	The system shall prevent DoS attack or unusual network traffic.
24	The cloud server service provider shall be well-known in the market.
25	The network service provider shall be well-known in the market.
26	The HD camera shall use 4G LTE technology for live transmission of Soccerwatch.
27	The system shall use Real-Time Transfer Protocol (RTP) and Real-Time Streaming Protocol (RTSP) for secure live transmission.
28	The system shall monitor, record and report an unauthorized activity on server.
29	The system shall prevent any malicious attempts (SQL injection or Man-in-the-Middle attack) on server, in this case the system shall notify to admin.
30	The system shall ensure that the Soccerwatch web browser shall use HTTPS.
31	The system shall not allow unauthorized person to monitor network communication.
32	The system shall ensure that all communication from LTE antenna to public server shall be encrypted.
	APPLICATION LAYER
33	The system shall notify to admin, in case of any malicious activity (e.g. data manipulation/tampering)
34	The system shall have a strong authentication for user.
35	The system shall not disclose Soccerwatch sensitive information to unauthorized person or other organization.
36	The system shall not disclose club member's sensitive information to unauthorized person.
37	The system shall have a strong authentication for club members.
38	The system shall not allow unauthorized persons to access club member's information.
39	The system shall not allow unauthorized person to access local server to avoid data manipulation.
40	The system shall ensure that the software updates in user application should not have any negative (vulnerability) effects on connectivity.
41	The system shall have load balancer to handle excessive request from clients.
42	The system shall add multiple nodes to load balancer to share request load.

6.3 Comparative Analysis of Frameworks

In this work, we present a comparative analysis with the most commonly-used security requirements frameworks (Chapter 4) of our proposed CPS framework activities. The proposed framework activities are defined as criteria to be used as a reference when comparing with existing frameworks. This comparative analysis helps us to determine the strengths and weaknesses of each framework against our proposed SRE framework. Table 6.3 shows the comparative analysis of different security requirements frameworks.

Our findings from this comparative analysis indicate that the benefits of these frameworks are limited to the realm of software, and at some point, to supporting the computer hardware. This, together with lack of any activities to deal with problems specific to the physical layer may result in the development of unsecured cyber-physical systems. Unfortunately, none of these frameworks focuses on addressing the new problem of cyber-physical systems, which result from the difference in architecture between classical and cyber-physical systems. As said before, the most prominent difference is the addition of the physical environment as an integral part of the CPS, necessitating a state of continuous communication with the rest of the system. In this regard, sensors to monitor the real world and much more extensive communication networks are of paramount significance.

Furthermore, the diversity of cyber-physical systems forces the developer to take into consideration details of the security aspects of sensors, receivers, data processors, and communicators, not just the general software security aspect which is addressed in existing security requirements frameworks. Our proposed SRE framework overcomes the issue of security requirements elicitation for heterogeneous CPS components. The proposed framework supports the elicitation of security requirements while considering sensor, receiver protocol, network channel issues, along with software aspects. Every activity in the framework contributes to determining the security

requirements for CPS. The framework has also contributed to the identification of new threats for CPS that are not identified in existing frameworks. For example, threat on sensor data diversion to unknown server and DoS attack on camera by laser light.

Table 6.3 Comparative analysis

CPS Framework Activities	**Frameworks**						
	SQUARE [123]	***MS SDL*** [168]	***UMLsec*** [7]	***Secure Tropos*** [172]	***CLASP*** [160]	***SREP*** [141]	***CORAS*** [159]
A1		**x**	**x**		**x**	**x**	**x**
A2	**x**		**x**	**x**		**x**	
A3	**x**	**x**	**x**		**x**	**x**	**x**
A4							
A5							
A6							
A7	**x**	**x**			**x**	**x**	**x**
A8	**x**	**x**	**x**	**x**	**x**	**x**	

Our recognized activities (A1 to A8) are designed to focus on all three layers (i.e. application layer, network layer and physical layer) of CPS, while other frameworks dealt exclusively with the application layer. This may be in part due to the extensively documented security measures for the network layer already in existence, and a general lack of attention to physical layer security in the past. However, this still leaves a hole in the entire procedure, which needs to be filled to avoid lapses in cyber-physical-system security. This framework, by explicitly taking into account, all three layers attempts to do just that.

Given the fact that CPS are tightly coupled to the physical environment, and interact with it directly by means of sensors, actuators and gateways and that they are more

vulnerable to external threats from the physical environment than to cyber ones, it is imperative for any CPS security framework to address the security of the physical layer in detail. In this case, merely relying on standalone physical layer protections provided by manufacturers does not provide holistic security from all realms of attack in light of their cross-layer functions (as detailed in Chapter 2) in the CPS. Therefore, it is very important to explore some new security requirements frameworks, especially for CPS. More specifically, in our proposed SRE framework, we recognized three major activities (i.e. A4 to A6) which support the analyst to understand the network and physical environment and help to determine the security requirements for CPS. These activities are not explicitly mentioned in other frameworks. Ignoring the activities from A4 to A6 means ignoring the security requirements of the physical layer. While previously available frameworks when applied to CPS may be used to also touch upon elements of the physical layer and hardware, they are nowhere near enough to deal with hundreds of remotely placed and environmentally exposed physical components that CPS involve. In addition, even where previous frameworks discuss secure communication protocols, those same internal network communication protocols are not necessarily optimal or even secure to operate with sensors placed in the public sphere, and thus, A6 gains distinction as a necessary and separate activity from A4. This is particularly true given the weakness of most current day communication protocols for sensor network applications, a field that demands more research given the needs of modern CPS.

In our framework, we describe in detail the activities that help to elicit the security requirements for CPS. While other frameworks also present a number of activities, some of them of a similar nature to those we present, they focus for the most part on the software side and do not explicitly mention the other layers of CPS, which are critical to determine the security of CPS. For example, the original SQUARE framework does not determine the assets, and Secure Tropos methodology also does not look for assets. However, the other frameworks address the assets directly, but

their use has conventionally been limited only to software-based assets, and no explicit mention of physical assets is made. Similarly, while security goals such as availability are discussed by many frameworks, it is addressed from the perspective of availability of the application, web-service or server, but none mandate the need for availability of sensor nodes that maybe arbitrarily distant from the rest of the system. The same can be said for the entirely new dimension of threats that come into play in the physical layer such as damage, malfunction or malicious manipulation, particularly where sensor and actuator nodes are placed unattended and distributed over a wide region. It must be pointed out at this point that most of these frameworks were originally designed to cater to the needs of software systems at a time when CPS were still relatively obscure, and thus these frameworks were designed under no pressing need to consider elements of the physical layer at all, or even of the network layer beyond a limited scope that was relevant for software systems prevalent at the time.

Risk assessment is often considered a key part of security requirements engineering. That's why, the risk assessment methods like SQUARE, MS SDL, CLASP, SREP and CORAS focus on this main activity, which they perform satisfactorily if given an appropriate set of inputs like assets, threats, etc. Specially, CORAS is known for its elaborate risk assessment capabilities and they have provided the detail methodology of risk assessment. In A7, we have extended the risk assessment methodology proposed by NIST [212] (SQUARE has adopted this methodology in its implementation) and adapted it to be used with CPS.

Additionally, our proposed framework simplifies the process for an analyst by presenting the activities in the form of a checklist from which relevant items may be selected and by identifying the threats and security requirements through the technique of misuse case, which has shown to be capable of eliciting security requirements in a straightforward manner. While the other frameworks provide detailed methodology for executing the core activities, they are not briefly summed

up in a single checklist as we have presented, and at times, can be difficult to understand. We evaluated our framework through two case studies and determined the security requirements for their respective CPS. It was observed in initial stages of application of our framework to the case study 'Soccerwatch' that the conventional approach of open-discussion with stakeholders showed very slow progress, and the resulting development of checklists for subsequent meetings made communication much more organized, productive and comprehensible for stakeholders who were unfamiliar with the security requirements engineering process. Our proposed framework is a concise, independent and self-sufficient approach to security requirements engineering for CPS. It contains not only all the guidelines for the process, but also contains detailed checklists that make identifying various relevant security elements not much more than a lookup job, except where special application specific elements may need to be identified on one's own.

This comparative analysis helps to provide us a frame of reference for judging existing SRE frameworks, and to better highlight the contribution to judge the effectiveness of our proposed security requirements engineering framework for CPS. Here we have taken a look at the strengths and weaknesses of each framework. Our findings indicate that none of the frameworks fulfil all the desired functionality expected of a secure CPS. This appears to be primarily because most security requirement frameworks in use today have been designed to work with software systems, the needs of which form only a sub-set of the needs of a security requirements framework for cyber-physical systems. As CPS started to become widespread, it has been attempted to apply existing security requirements frameworks to CPS, but they have been found to be inadequate for CPS purposes [181]. This comparison with our own framework further cements the idea that existing security requirements frameworks cannot be employed to satisfactorily guarantee CPS security, and thus justifies the need and utility of such a dedicated framework, which we have attempted to deliver in this work.

6.4 Chapter Summary

In this chapter, we presented two case studies describing how the CPS framework was used to elicit security requirements for a smart car parking system and Soccerwatch, applied at a real world scenario. We described all the activities and described how the framework is to be applied on a case study to measure the effectiveness of the framework. Our framework allows us to systematically determine the security requirements. We evaluated our framework by applying it to two case studies. By applying first to the case study of the smart car parking system, we determined 43 security requirements and for the second case study of Soccerwatch, we were able to determined 42 security requirements. Such, we were able to evaluate the effectiveness of our proposed framework. Furthermore, we also compared our proposed CPS framework with other existing SRE frameworks. Our findings from this comparison survey indicate that none of the frameworks performs all the required activities for secure cyber-physical systems. The result would provide great support in this research direction.

CHAPTER 7

Conclusions and Future Work

This chapter concludes the thesis by giving a summary of its main content in section 7.1 and recapping the basic research questions posed in chapter 1, and reviewing how we have addressed these questions through the course of this research in section 7.2. We provide possibilities for future work with the context of this thesis in section 7.3.

7.1 Summary

Security requirements are a significant part of cyber-physical systems, but there is a lack of processes to develop secure systems. Many security requirements methodologies are in use today, but these are limited only to the realm of software, and none supports cyber-physical systems. This is particularly significant given the uniquely different needs of cyber-physical systems, due to the addition of the physical layer. This opens up many new avenues of attack and makes it necessary to be aware of the new threats to CPS that have emerged. We have detailed the types of security challenges that CPS face and what the most common types of misactors tend to be. We have outlined the existing well-known security requirements frameworks for software systems, and compared their strengths and weaknesses. We also conducted a systematic mapping study for security requirements engineering for cyber-physical systems and presented our results, which offer a valuable contribution to the literature. In this thesis, our main contribution is to provide a comprehensive security requirements engineering framework for cyber-physical systems that can offer complete guidelines for practitioners and researchers to determine security requirements. The novelty of such an implementation is that such a security requirements engineering framework for CPS at this scale has not been significantly reported in the literature. The proposed CPS framework is designed to analyse cyber-physical systems, its architecture and environment to identify security requirements throughout the requirements engineering phase. This is done through eight essential

activities defined in the framework. To facilitate the elicitation of security requirements, we have also developed a SRE Tool to guide and formalize the implementation of the eight activities of the framework. To evaluate the CPS framework, we have applied our approach to two case studies, namely, a smart car parking system and a real-world implementation with our industrial partner, Soccerwatch. We obtained promising results from both case studies, having elicited 43 and 42 security requirements respectively, using the SRE tool designed to accompany our proposed CPS framework. Furthermore, our proposed security requirements engineering framework is compared with other existing software security frameworks. It was found that none of the software security frameworks implements all the essential activities for the development of secure CPS defined in our SRE framework.

The research community can benefit greatly from this framework. Every activity in the framework contributes to determining the security requirements for CPS. The framework has also contributed to the identification of new threats for CPS that were not identified earlier in this field. Recently, cyber threats are on the rise, and in order to secure cyber physical systems from being at risk, and hence minimize the economic and even life-threatening consequences, this framework has the potential to contribute significantly. For the researcher, the framework will help them to explore security threats of CPS in more detail. Being quite broad in its scope, the framework is easily adaptable to various applications according to their needs.

The findings of this research will be of great benefit to practitioners and researchers who play an important role in the development of security requirements engineering for CPS. The increased demand for security in organizations justifies the need for a security requirements engineering framework. Organizations that apply the proposed framework derived from the results of this research will be better positioned to explore security requirements in the early phases of system development and be assured of an uncompromised system of security.

7.2 Addressing Research Questions

At the beginning of this work, we outlined the basic problem statement, from which we extracted five fundamental research questions. Over the course of this thesis, we endeavoured to find satisfactory answers to these questions and present them systematically. Here, we will briefly go over those fundamental questions and how we have addressed them in this study.

In chapter 2, we presented a detailed outline of the kinds of threats that CPS are susceptible to. This was done by analysing a structure of CPS functional nodes and identifying all potential points of attack, along with associated actions that would compromise security. It was found that much of the important threats for CPS were those that targeted the physical layer, as it is the most vulnerable. This threat analysis addresses *RQ1.'Which security threats are most important for cyber-physical systems?'.*

In chapter 4, we introduced the existing security requirements engineering frameworks for software systems, namely, SQUARE, Microsoft SDL, UMLsec, Secure Tropos, SREP and CORAS. We compared their domains and functionalities, showing which security elements they each addressed, along with analysing their respective strengths and weaknesses. This review of existing frameworks answers *RQ2. 'What are the existing security requirements engineering frameworks to specify the security of software?'* and *RQ3. 'Do existing security requirements frameworks fulfil the needs of cyber-physical systems?'* In chapter 6, we further compared existing security requirements engineering frameworks with our proposed framework on the basis of the eight essential activities we defined, and found that none of the existing frameworks employed all of these activities.

In chapter 5, we proposed our CPS framework for eliciting security requirements. The penultimate activity involved a detailed risk assessment step for analysing assets and their associated levels of risk. We developed a CPS Risk Matrix for this risk assessment that involved the threat likelihood and impact values are used to calculate the risk to

each respective asset. This proposed technique addresses *RQ4. 'Which risk assessment technique can be utilized for the security requirements framework of CPS?'*.

In chapter 6, we evaluated our proposed CPS framework by applying it to two case studies. By applying first to the case study of the smart car parking system, we elicited 43 security requirements and for the second case study of Soccerwatch, we are able to elicited 42 security requirements. Such, we are able to evaluate the effectiveness of our proposed CPS framework. This forms the answer to *RQ 5. 'How effective is the proposed SRE framework in eliciting of security requirements for cyber-physical systems?'*.

7.3 Future Work

Though the problem statement and associated research questions are quite vast, we have attempted in this work to provide satisfactory answers to these questions. However, in the broad domain of security for cyber-physical systems, there is still much work to be done, both in terms of security requirements as well as actual security implementation. In this section, we have highlighted some of the important avenues of further research that have been indicated either directly or indirectly from the conclusions of this research, which are as follows:

7.3.1 Threat Modelling

The proposed CPS framework can be used for further research development into meta-models, threat modelling and security design to enhance the security methods. For the researcher, this will help them to explore security threats to cyber-physical systems in more detail. The proposed CPS framework can be applied in various contexts and is flexible enough to be adapted according to the need in each respective case. The proposed CPS framework can be applied in the design phase to extend the concepts of attack tree interaction with CPS heterogeneous components. The proposed framework is an initial step that can be used not only to determine security requirements, but also to open up new research directions. The framework has been

used to identify new threats for cyber-physical systems that were not explicitly stated earlier in this field, and so motivates further research with regards to these threats.

7.3.2 Intrusion Detection Systems

Intrusion Detection Systems (IDS) are tools to detect threats and alert the system about malicious system behaviour. IDS contribute significantly to system security in run-time after system deployment. However, conventional intrusion detection systems are not designed for cyber-physical systems as they usually operate in the domains of the network (Network-based Intrusion Detection System (NIDS)) or application layer (Host-based Intrusion Detection System (HIDS)). This has left the physical layer unguarded and susceptible to malicious acts that may result in undetected intrusions into the system. For effective security of cyber-physical systems in the future, it is imperative to develop IDS that incorporate security for the physical layer as well as network and application layers.

7.3.3 Security Standards

There is a need to consolidate universal security standards that are both up-to-date and approved by competent authority. This set of security standards should transcend specificity to any one layer or portion of the CPS, and should offer a holistic security framework that ensures protection against malicious agents. They should provide a coordinated approach to security related interactions within the CPS, particularly between layers. Standardising these security standards will make it easier to diagnose any breach in security and be able to determine its root cause and operating procedures to prevent future recurrences.

Some pertinent examples include developing a secure communication protocol for cyber-physical systems. Given the wide array of sensor and network communication channels, it has become difficult to ensure uniform security over all data communication, which may lead to greater breaches in security. Similarly, it is recommended to dedicate a new, uniform and secure language and environment for

programming cyber-physical systems that possesses features to make it robust to malicious agents. In addition, a standardised family of security software that has access to a common database containing latest information about security threats to CPS should be established.

References

[1] S. Rehman and V. Gruhn, "Recommended Architecture for Car Parking Management System based on Cyber-Physical System," in *Proceedings of the International Conference on Engineering & MIS*, Monastir, 2017.

[2] C. Kalloniatis, E. Kavakli and S. Gritzalis, "Addressing privacy requirements in system design: the PriS method," in *Requirements Eng. 13, 241–255.*, 2008.

[3] L. Zhang, "A framework to specify big data driven complex cyber physical control systems," in *Proceedings of the 2014 IEEE International Conference on 2014 Information and Automation (ICIA), pp. 548–553*, Hailar, China, 28–30 July 2014.

[4] S. Rehman, A. Hark and V. Gruhn, "A framework to handle big data for cyber-physical systems," in *8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (pp. 72-78). IEEE*, 2017.

[5] S. Rehman, M. Prehn and V. Gruhn, "SmartChair: A Realization of Smart Health Care System based on Cyber-Physical Systems," in *Proceedings of the Fourth International Conference on Engineering & MIS 2018 (p. 20). ACM*, 2018.

[6] A. Wong, "Cyber Security Threats Challenges Opportunities," ACS, 2016.

[7] J. Jürjens, Secure systems development with UML, Springer Science & Business Media, 2005.

[8] G. Elahi, E. Yu and Zannone, "A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities," in *Requirements engineering, 15(1), pp.41-62*, 2010.

[9] P. Salini and S. Kanmani, "Survey and analysis on security requirements engineering," in *Computers & Electrical Engineering, 38(6), 1785-1797.*, 2012.

[10] K. Beckers, S. Faßbender, D. Hatebur, M. Heisel and I. Côté, "Common criteria compliant software development (CC-CASD)," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing,; pp. 1298–1304.*, Coimbra, Portugal, 18–22 March 2013.

[11] M. Lund, B. Solhaug and K. Stølen, "Model-driven risk analysis: The CORAS approach," in *Springer Science & Business Media*, New York, NY, USA, 2010.

[12] "CC: ISO/IEC 15408 Information Technology—Security Technology—Evaluation Criteria for IT Security V2.1.," December 2009. [Online]. Available: https://www.iso.org/standard/50341.html.

[13] V. Lamsweerde, "A. Goal-Oriented Requirements Engineering: A Guided Tour," in *Proceedings of the RE'01: 5th International Symposium on Requirements Engineering,*, Toronto, ON, Canada, 27–31 August 2001.

[14] J. Jürjens, "UMLsec: Extending UML for secure systems development," in *Proceedings of the «UML» pp. 1–9. The Unified Modeling Language*, Dresden, Germany, 2002.

[15] P. Leitão, A. Colombo and S. Karnouskos, "Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges," in *Computers in Industry, 81, pp.11-25.*, 2016.

[16] F. Cleveland, "Cyber Security Issues for Advanced Metering Infrastructure (AMI)," in *Proceeding of the IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical energy in the 21st Century,*, Pittsburgh, PA, USA,, 20–24 July 2008.

[17] A. Humayed, J. Lin, F. Li and B. Luo, "Cyber-physical systems security—A survey," in *IEEE Internet of Things Journal, 4(6), 1802-1831.*, 2017.

[18] Y. Ashibani and Q. H. Mahmoud, "Cyber physical systems security. Analysis, challenges and solutions," in *Computers & Security 68, pp. 81–97. DOI: 10.1016/j.cose.2017.04.005.*, 2017.

[19] Y. Gao, Y. Peng, F. Xie, W. Zhao, D. Wang, X. Han, T. Lu and Z. Li, "Analysis of security threats and vulnerability for cyber-physical systems," in *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology (pp. 50-55). IEEE*, 2013.

[20] R. Anderson, Security engineering: A guide to building dependable distributed systems, John Wiley & Sons., 2010.

[21] G. McGraw, Software security: building security in (Vol. 1)., Addison-Wesley Professional., 2006.

[22] M. U. A. Khan and M. Zulkernine, "On selecting appropriate development processes and requirements engineering methods for secure software," in *Computer Software and Applications Conference, 2009. COMPSAC'09. 33rd Annual IEEE International (Vol. 2, pp. 353-358). IEEE*, 2009.

[23] S. Rehman, M. Ceglia, S. Siddiqui and V. Gruhn, "Towards an Importance of Security for Cyber-Physical Systems/Internet-of-Things," in *The 8th ACM International Conference on Software and Information Engineering (ICSIE 2019). ACM*, 2019.

[24] S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security," in *IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society (pp. 4490-4494). IEEE.*, 2011.

[25] J. Slay and M. Miller, "Lessons learned from the maroochy water breach," in *Proceedings of the In Proceedings of the International Conference on Critical Infrastructure Protection, Boston, MA, USA, 19 March 2007; pp. 73–82.*, 2007.

[26] L. Wells, J. Camelio, C. Williams and J. White, "Cyber-physical security challenges in manufacturing systems," in *Manufacturing Letters, 2(2), pp.74-77.*, 2014.

[27] S. Rehman, D. Ceglia, M. and V. Gruhn, "Analysing Security Threats for Cyber-Physical Systems," in *Future of Information and Communication Conference (pp. 1095-1105). Springer, Cham.*, 2019.

[28] S. Rehman, A. Iannella and V. Gruhn, "A Security Based Reference Architecture for Cyber-Physical Systems," in *International Conference on Applied Informatics (pp. 157-169). Springer, Cham.*, 2018.

[29] L. Gurgen, O. Gunalp, Y. Benazzouz and M. Gallissot, "Self-aware cyber-physical systems and applications in smart buildings and cities," in *Design, Automation & Test in Europe Conference & Exhibition (DATE) (pp. 1149-1154). IEEE.*, 2013.

[30] R. Rajkumar, I. Lee, L. Sha and J. Stankovic, "Cyber-physical systems: the next computing revolution," in *Design Automation Conference (DAC), 2010 47th ACM/IEEE (pp. 731-736). IEEE.*, 2010.

[31] L. Da Xu, W. He and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics, 10(4), 2233-2243.*, 2014.

[32] J. Gubbi, R. Buyya and S. P. M. Marusic, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems, 29(7), pp.1645-1660,* 2013.

[33] J. Wan, H. Yan, H. Suo and F. Li, "Advances in Cyber-Physical Systems Research.," *KSII Transactions on Internet & Information Systems,*, p. 5(11), 2011.

[34] M. E. Brak, S. E. Brak, M. Essaaidi and D. Benhaddou, "Wireless Sensor Network applications in smart grid," in *International Renewable and Sustainable Energy Conference (IRSEC) (pp. 587-592). IEEE.*, 2014.

[35] B. Bordel, R. Alcarria, T. Robles and D. Martín, "Cyber–physical systems: Extending pervasive sensing from control theory to the Internet of Things," in *Pervasive and mobile computing, 40, 156-184.*, 2017.

[36] D. Estrin, D. Culler, K. Pister and G. Sukhatme, "Connecting the physical world with pervasive networks," in *IEEE pervasive computing, 1(1), 59-69.*, 2002.

[37] L. M. Oliveira and J. J. Rodrigues, "Wireless Sensor Networks: A Survey on Environmental Monitoring," in *JCM, 6(2), 143-151.*, 2011.

[38] S. Wang, J. Wan, D. Li and C. Zhang, "Implementing smart factory of industrie 4.0: an outlook," in *International Journal of Distributed Sensor Networks, 12(1), 3159805.*, 2016.

[39] W. Dargie and M. Zimmerling, "Wireless sensor networks in the context of developing countries," in *In IFIP World IT Forum (WITFOR).*, 2007.

[40] N. Jazdi, "Cyber physical systems in the context of Industry 4.0," in *2014 IEEE international conference on automation, quality and testing, robotics (pp. 1-4). IEEE.*, 2014.

[41] Y. Tan, S. Goddard and L. Perez, "A prototype architecture for cyber-physical systems," in *ACM Sigbed Review, 5(1), p.26*, 2008.

[42] S. Han, M. Xie, H. Chen and Y. Ling, "Intrusion detection in cyber-physical systems: Techniques and challenges," in *IEEE systems journal, 8(4), pp.1052-1062*, 2014.

[43] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang and W. Zhao, "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications," in *IEEE Internet of Things Journal, 4(5), 1125-1142*, 2017.

[44] A. B. Sharma, F. Ivančić, A. Niculescu-Mizil, H. Chen and G. Jiang, "Modeling and analytics for cyber-physical systems in the age of big data," in *ACM SIGMETRICS Performance Evaluation Review, 41(4), 74-77.*, 2014.

[45] L. Zhang, "Designing big data driven cyber physical systems based on AADL," in *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on (pp. 3072-3077). IEEE.*, 2014.

[46] J. Shi, J. Wan, H. Yan and H. Suo, "A survey of cyber-physical systems," in *Wireless Communications and Signal Processing (WCSP), 2011 International Conference on (pp. 1-6). IEEE.*, 2011.

[47] E. A. Lee, "Cyber physical systems: Design challenges," in *11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC) (pp. 363-369). IEEE.*, 2008.

[48] L. Zhang, "Modeling large scale complex cyber physical control systems based on system of systems engineering approach," in *20th International Conference on Automation and Computing (pp. 55-60). IEEE.*, 2014.

[49] Y. Guan and X. Ge, "Distributed attack detection and secure estimation of networked cyber-physical systems against false data injection attacks and jamming attacks," in *IEEE Transactions on Signal and Information Processing over Networks, 4(1), 48-59.*, 2017.

[50] Y. Peng, T. Lu, J. Liu, Y. Gao, X. Guo and F. Xie, "Cyber-physical System Risk Assessment," *2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing.*, 2013.

[51] T. Lu, J. Lin, L. Zhao, Y. Li and Y. Peng, "A Security Architecture in Cyber-Physical Systems. Security Theories, Analysis, Simulation and Application Fields," *IJSIA (International Journal of Security and Its Applications) 9 (7),* 2015.

[52] S. Tan, D. De, W. Z. Song, J. Yang and S. K. Das, "Survey of security advances in smart grid: A data driven approach," in *IEEE Communications Surveys & Tutorials, 19(1), 397-422.*, 2017.

[53] E. K. Wang, Y. Ye, X. Xu, S. M. Yiu, L. C. K. Hui and K. P. Chow, "Security Issues and Challenges for Cyber Physical System," in *IEEE/ACM Int'l Conference on Cyber, Physical and Social Computing (CPSCom). Hangzhou, China, 18.12.2010 - 20.12.2010: IEEE, pp. 733–738.*, 2010.

[54] H. Song, G. A. Fink and S. Jeschke, "Security and Privacy in Cyber-Physical Systems," in *Chichester, UK: Wiley.*, 2017.

[55] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks. Attacks and countermeasures.," in *International Workshop on Sensor Network Protocols and Applications. Piscataway, N.J: Institute of Electrical and Electronics Engineers, pp. 113–127.*, 2003.

[56] E. J. Yoon and I. S. Jeon, "An efficient and secure Diffie–Hellman key agreement protocol based on Chebyshev chaotic map," *Communications in Nonlinear Science and Numerical Simulation, 16(6), 2383-2389.*, 2011.

[57] J. S. Coron, A. Joux, I. Kizhvatov, D. Naccache and P. Paillier, "Fault attacks on RSA signatures with partially unknown messages," *Cryptographic Hardware and Embedded Systems-CHES 2009 (pp. 444-456). Springer, Berlin, Heidelberg.*, 2009.

[58] A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks," in *Proceedings of the 7th international conference on Information processing in sensor networks (pp. 245-256). IEEE Computer Society.*, 2008.

[59] A. S. Wander, N. Gura, H. Eberle, V. Gupta and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on (pp. 324-328). IEEE.*, 2005.

[60] R. Bhanot and R. Hans, "A review and comparative analysis of various encryption algorithms," in *International Journal of Security and Its Applications, 9(4), 289-306.*, 2015.

[61] G. McGraw, J. H. Allen, N. Mead, R. J. Ellison and S. Barnum, "Software Security Engineering: A Guide for Project Managers," *CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.*, 2013.

[62] C. B. Haley, J. D. Moffett, R. Laney and B. Nuseibeh, "A framework for security requirements engineering," in *Proceedings of the 2006 international workshop on Software engineering for secure systems (pp. 35-42). ACM.*, 2006.

[63] M. Ramachandran, "Software security requirements management as an emerging cloud computing service," *Int. J. Inf. Manag. 2016, 36, 580–590.*, 2016.

[64] N. Mead, "How to Compare the Security Quality Requirements Engineering (SQUARE) Method with Other Methods (No. CMU/SEI-2007-TN-021)," *Carnegie Mellon University Software Engineering Institute: Pittsburgh, PA*, 2007.

[65] N. Subramanian and J. Zalewski, "Quantitative assessment of safety and security of system architectures for cyber-physical systems using the NFR approach," *IEEE Syst. J. 2016, 10, 397–409.*, 2016.

[66] X. Yu and Y. Xue, "Smart grids: A cyber–physical systems perspective," in *Proceedings of the IEEE, 104(5), 1058-1070.*, 2016.

[67] M. Abomhara, "Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks," in *Journal of Cyber Security and Mobility, 4(1), 65-88.*, 2015.

[68] A. Nourian and S. Madnick, "A systems theoretic approach to the security threats in cyber physical systems applied to stuxnet," in *IEEE Transactions on Dependable and Secure Computing, 15(1), 2-13.*, 2015.

[69] J. Weiss, "Industrial Control System (ICS) cyber security for water and wastewater systems," in *Securing Water and Wastewater Systems (pp. 87-105). Springer, Cham.*, 2014.

[70] C. Wilson, "Cyber threats to critical information infrastructure," in *Cyberterrorism (pp. 123-136). Springer, New York, NY.*, 2014.

[71] T. Rid and B. Buchanan, "Attributing cyber attacks," in *Journal of Strategic Studies, 38(1-2), 4-37.*, 2015.

[72] G. Kumar and K. Kumar, "Network security–an updated perspective," in *Systems Science & Control Engineering: An Open Access Journal, 2(1), 325-334.*, 2014.

[73] W. Ashford, "On-demand service aims to cut cost of fixing software security flaws," in *http://www.computerweekly.com/Articles/2009/07/14/236875/on-demand-service-aims*, 2009.

[74] S. Rehman and M. Khan, "Security and Reliability Requirements for a Virtual Classroom," in *Proced. Comput. Sci. 2016, 94, 447–452.*, 2016.

[75] R. Robles and T. Kim, "Applications, Systems and Methods in Smart Home Technology: A review," *Int. J. Adv. Sci. Technol. 2010, 15, 37–48.*, 2010.

[76] M. Shahzad, M. Shafiq and A. Liu, "A large scale exploratory analysis of software vulnerability life cycles," in *Proceedings of the 34th International Conference on Software Engineering, Zurich, Switzerland, 2–9 June 2012; pp. 771–781.*, 2012.

[77] M. Almorsy, J. Grundy and I. Müller, "An analysis of the cloud computing security problem," in *arXiv 2016, arXiv:1609.01107.*, 2016.

[78] M. Khan and M. Zulkernine, "Quantifying security in secure software development phases," in *Proceedings of the 2008 32nd Annual IEEE International Computer Software and Applications Conference, Turku, Finland, 28 July–1 August 2008; pp. 955–960.*, 2008.

[79] I. Sommerville, D. Cliff, R. Calinescu, J. Keen, T. Kelly, M. Kwiatkowska and R. Paige, "Large-scale complex IT systems," in *arXiv preprint arXiv:1109.3444.*, 2011.

[80] A. Barabanov, A. Markov, A. Fadin, V. Tsirlov and I. Shakhalov, "Synthesis of secure software development controls," in *Proceedings of the 8th International Conference on Security of Information and Networks (pp. 93-97). ACM.*, 2015.

[81] A. Perrig, J. Stankovic and D. Wagner, "Security in wireless sensor networks. Commun," in *ACM 2004, 47, 53.*, 2004.

[82] Q. Yan, F. Yu, Q. Gong and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," in *IEEE Commun. Surv. Tutor. 2016; 18, 602–622.*, 2016.

[83] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen and D. E. Culler, "SPINS: Security protocols for sensor networks," in *Wireless networks, 8(5), 521-534.*, 2002.

[84] M. Raya and J. P. Hubaux, "Securing vehicular ad hoc networks," in *Journal of computer security, 15(1), 39-68.*, 2007.

[85] J. Åkerberg, M. Gidlund and M. Björkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *9th IEEE International Conference on Industrial Informatics (pp. 410-415). IEEE.*, 2011.

[86] G. Ericsson, "Cyber security and power system communication essential parts of a smart grid infrastructure," *IEEE Trans. Power Deliv. 2010, 25, 1501–1507.*, 2010.

[87] S. Sridhar and G. Manimaran, "Data integrity attacks and their impacts on SCADA control system," in *Proceedings of the IEEE PES General Meeting, Providence, RI, USA, 25–29 July 2010; pp. 1–6.*, 2010.

[88] G. Howser, "Using information flow methods to secure cyber-physical systems," in *International Conference on Critical Infrastructure Protection (pp. 185-205). Springer, Cham.*, 2015.

[89] S.-R. Oh and Y.-G. Kim, "Security Requirements Analysis for the IoT," in *Proceedings of the 2017 International Conference on Platform Technology and Service (PlatCon), Busan, South Korea, 13–15 February 2017; pp. 1–6.*, 2017.

[90] D. Gollmann and M. Krotofil, "Cyber-physical systems security," in *The New Codebreakers (pp. 195-204). Springer, Berlin, Heidelberg.*, 2016.

[91] S. Tedeschi, C. Emmanouilidis, J. Mehnen and R. Roy, "A design approach to IoT endpoint security for production machinery monitoring," in *Sensors, 19(10), 2355.*, 2019.

[92] R. Alguliyev, Y. Imamverdiyev and L. Sukhostat, "Cyber-physical systems and their security issues," in *Computers in Industry, 100, 212-223.*, 2018.

[93] D. Nunes, J. S. Silva and F. Boavida, "A Practical Introduction to Human-in-the-loop Cyber-physical Systems," in *John Wiley & Sons, Incorporated.*, 2018.

[94] S. Mohan, S. Bak, E. Betti, H. Yun, L. Sha and M. Caccamo, "S3A: Secure system simplex architecture for enhanced security and robustness of cyber-physical systems," in *Proceedings of the 2nd ACM international conference on High confidence networked systems (pp. 65-74). ACM.*, 2013.

[95] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu and M. Galloway, "A survey of key management schemes in wireless sensor networks," *Computer communications, 30(11-12), 2314-2341.*, 2007.

[96] B. Zhu, A. Joseph and S. Sastry, "A taxonomy of cyber-attacks on SCADA systems. In Proceedings of the Internet of things (iThings/CPSCom)," in *International Conference on and 4th International Conference on Cyber, Physical and Social Computing, Washington, DC, USA, 9–22 October 2011; pp. 380–388*, 2011.

[97] A. Mosenia and N. K. Jha, "A comprehensive study of security of internet-of-things," in *IEEE Transactions on Emerging Topics in Computing, 5(4), 586-602.*, 2017.

[98] W. Ao, Y. Song and C. Wen, "Adaptive cyber-physical system attack detection and reconstruction with application to power systems," in *IET Control Theory & Applications, 10(12), 1458-1468.*, 2016.

[99] K. Manandhar, X. Cao, F. Hu and Y. & Liu, "Detection of faults and attacks including false data injection attack in smart grid using kalman filter," in *IEEE transactions on control of network systems, 1(4), 370-379.*, 2014.

[100] M. Burmester, E. Magkos and V. Chrissikopoulos, "Modeling security in cyber–physical systems," *Int. J. Crit. Infrastruct. Protect. 2012, 5, 118–126.*, 2012.

[101] D. Papp, Z. Ma and L. Buttyan, "Embedded systems security: Threats, vulnerabilities, and attack taxonomy," in *13th Annual Conference on Privacy, Security and Trust (PST) (pp. 145-152). IEEE.*, 2015.

[102] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu and D. Qiu, "Security of the Internet of Things: perspectives and challenges," in *Wireless Networks, 20(8), 2481-2501.*, 2014.

[103] K. Zeng, "Physical layer key generation in wireless networks: challenges and opportunities," *IEEE Commun. Mag. 2015, 53, 33–39,* 2015.

[104] T. Lu, J. Zhao, L. Zhao, Y. Li and X. Zhang, "Towards a framework for assuring cyber physical system security," in *International Journal of Security and Its Applications, 9(3), 25-40.*, 2015.

[105] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Federated Conference on Computer Science and Information Systems (FedCSIS), Warsaw, Poland, 7–10 September 2014; pp. 1–8.*, 2014.

[106] R. Di Pietro, S. Guarino, N. V. Verde and J. Domingo-Ferrer, "Security in wireless ad-hoc networks–a survey," *Computer Communications, 51, 1-20.*, 2014.

[107] H. Orojloo and M. A. Azgomi, "A game-theoretic approach to model and quantify the security of cyber-physical systems," in *Computers in Industry, 88, 44-57.*, 2017.

[108] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer and Y. Weiss, "Andromaly: a behavioral malware detection framework for android devices," in *Journal of Intelligent Information Systems, 38(1), 161-190.*, 2012.

[109] M. Farooq, M. Waseem, A. Khairi and S. Mazhar, "A critical analysis on the security concerns of internet of things (IoT)," *Int. J. Comput. Appl. 2015, 111. Available online: http://www.pcporoje.com/filedata/592496.pdf (accessed on 10 July 2018),* 2018.

[110] N. Conteh and P. Schmick, "Cybersecurity: risks, vulnerabilities and countermeasures to prevent social engineering attacks," in *International Journal of Advanced Computer Research, 6(23), p.31.*, 2016.

[111] "https://www.bsi.bund.de/SharedDocs/Downloads/EN/.../threats_catalogue.pdf," [Online].

[112] J. McClean, C. Stull, C. Farrar and D. Mascareñas, "A preliminary cyber-physical security assessment of the robot operating system (ros)," in *Unmanned Systems Technology XV (Vol. 8741, p. 874110). International Society for Optics and Photonics*, 2013.

[113] Y. Wang, B. Ramamurthy, X. Zou and Y. Xue, "An efficient scheme for removing compromised sensor nodes from wireless sensor networks," in *Security and Communication Networks, 3(4), 320-333.*, 2010.

[114] M. K. Jain, "Wireless sensor networks: Security issues and challenges," in *International Journal of Computer and Information Technology, 2(1), 62-67*, 2011.

[115] C. W. Probst, J. Hunker, D. Gollmann and M. Bishop, "Aspects of insider threats. In Insider Threats in Cyber Security (pp. 1-15)," in *Springer, Boston, MA.*, 2010.

[116] A. Cárdenas, S. Amin, Z. Lin, Y. Huang, C. Huang and S. Sastry, "Attacks Against Process Control Systems: Risk Assessment, Detection, and Response," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security,*, 2011.

[117] M. McGuire and S. Dowling, "Cyber crime: A review of the evidence. Summary of key findings and implications," *Home Office Research report, 75.*, 2013.

[118] S. Rehman, V. Gruhn, S. Shafiq and I. Inayat, "A Systematic Mapping Study of Security Requirements Engineering for Cyber-Physical Systems," in *The 7th International Symposium on Security and Privacy on Internet of Things, (pp. 428-442) (Springer SpaCCS)*, 2018.

[119] L. D. Xu, E. L. Xu and L. Li, "Industry 4.0: state of the art and future trends," in *International Journal of Production Research, 56(8), 2941-2962.*, 2018.

[120] B. Fabian, S. Gürses, M. Heisel, T. Santen and H. Schmidt, "A comparison of security requirements engineering methods," in *Requirements engineering, 15(1), pp.7-40.*, 2010.

[121] D. Mellado, C. Blanco, L. E. Sánchez and E. Fernández-Medina, "A systematic review of security requirements engineering," in *Computer Standards & Interfaces, 32(4), 153-165.*, 2010.

[122] T. Gopal, M. Subbaraju, R. vivek Joshi and S. Dey, "MAR (S) 2: methodology to articulate the requirements for security in SCADA," in *Innovative Computing Technology (INTECH), 2014 Fourth International Conference on (pp. 103-108). IEEE*, 2014.

[123] N. Mead and T. Stehney, "Security Quality Requirements Engineering (SQUARE) Methodology," in *ACM: New York, NY, USA, 2005; Volume 30, pp. 1–7.*, 2005.

[124] K. Stolen, F. den Braber, T. Dimitrakos, R. Fredriksen, A. Gran B, H. Houmb S and O. Aagedal J, " Model-based risk assessment–the CORAS approach.," in *In Proceedings of the NIK (2002) Informatics Conference, Kongsberg, Norway, 25–27 November 2002*, 2002.

[125] P. Giorgini, H. Mouratidis and N. Zannone, "Modelling security and trust with secure tropos," in *Integrating Security and Software Engineering: Advances and Future Visions (pp. 160-189). IGI Global.*, 2007.

[126] S. Yahya, M. Kamalrudin and S. Sidek, "A review on tool supports for security requirements engineering," in *In Open Systems (ICOS), IEEE Conference on (pp. 190-194)*, 2013.

[127] I. A. Tondel, M. G. Jaatun and P. H. Meland, "Security requirements for the rest of us: A survey," in *IEEE software, 25(1).*, 2008.

[128] H. Yoo and T. Shon, "Challenges and research directions for heterogeneous cyber–physical system based on IEC 61850: Vulnerabilities, security requirements, and security architecture," in *Future Generation Computer Systems, 61, pp.128-136.*, 2016.

[129] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati and G. P. Hancke, "Smart grid technologies: Communication technologies and standards," in *IEEE transactions on Industrial informatics, 7(4), 529-539.*, 2011.

[130] K. Beckers, M. Heisel, B. Solhaug and K. Stølen, "ISMS-CORAS: A structured method for establishing an ISO 27001 compliant information security management system," in *Engineering Secure Future Internet Services and Systems (pp. 315-344). Springer*, 2014.

[131] W. Boehmer, "Appraisal of the effectiveness and efficiency of an information security management system based on ISO 27001," in *Second International Conference on Emerging Security Information, Systems and Technologies (pp. 224-231). IEEE*, 2008.

[132] N. Belloir, V. Chiprianov, M. Ahmad, M. Munier, L. Gallon and J. Bruel, "Using relax operators into an mde security requirement elicitation process for systems of systems," in *Proceedings of the 2014 European Conference on Software Architecture Workshops (p. 32). ACM*, 2014.

[133] Y. Liu, Y. Peng, B. Wang, S. Yao and Z. Liu, "Review on cyber-physical systems," in *IEEE/CAA Journal of Automatica Sinica, 4(1), pp.27-40.*, 2017.

[134] K. Petersen, R. Feldt, S. Mujtaba and M. Mattsson, "Systematic Mapping Studies in Software Engineering," in *EASE (Vol. 8, pp. 68-77).*, 2008.

[135] E. Paja, F. Dalpiaz and P. Giorgini, "Managing security requirements conflicts in socio-technical systems," in *International Conference on Conceptual Modeling (pp. 270-283). Springer, Berlin, Heidelberg.*, 2013.

[136] G. Wimmel and A. Wisspeintner, "Extended description techniques for security engineering," in *IFIP International Information Security Conference (pp. 469-485). Springer, Boston, MA.*, 2001.

[137] J. L. Vivas, J. A. Montenegro and J. López, "Towards a business process-driven framework for security engineering with the UML," in *International Conference on Information Security (pp. 381-395). Springer, Berlin, Heidelberg.*, 2003.

[138] T. Srivatanakul, J. A. Clark and F. Polack, "Effective security requirements analysis: Hazop and use cases," in *International Conference on Information Security (pp. 416-427). Springer, Berlin, Heidelberg.*, 2004.

[139] H. Abie, D. B. Aredo, T. Kristoffersen, S. Mazaher and T. Raguin, "Integrating a security requirement language with UML," in *International Conference on the Unified Modeling Language (pp. 350-364). Springer, Berlin, Heidelberg.*, 2004.

[140] P. Giorgini, F. Massacci and N. Zannone, "Security and trust requirements engineering," in *Foundations of Security Analysis and Design III (pp. 237-272). Springer, Berlin, Heidelberg.*, 2005.

[141] D. Mellado, E. Fernández-Medina and M. Piattini, "Applying a security requirements engineering process," in *European Symposium on Research in Computer Security (pp. 192-206). Springer, Berlin, Heidelberg.*, 2006.

[142] C. B. Haley, R. C. Laney, J. D. Moffett and B. Nuseibeh, "Using trust assumptions with security requirements," in *Requirements Engineering, 11(2), 138-151.*, 2006.

[143] V. Bryl, F. Massacci, J. Mylopoulos and N. Zannone, "Designing security requirements models through planning," in *International Conference on Advanced Information Systems Engineering (pp. 33-47). Springer, Berlin, Heidelberg.*, 2006.

[144] A. Herrmann and B. Paech, "MOQARE: Misuse-oriented quality requirements engineering," in *Requir. Eng., vol. 13, no. 1, pp. 73–86*, 2008.

[145] E. Moradian and A. Hkansson, "Controlling security of software development with multi-agent system," in *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6279 LNAI, no. PART 4, pp. 98–107*, 2010.

[146] R. Rieke, L. Coppolino, A. Hutchison, E. Prieto and C. Gaber, "Security and reliability requirements for advanced security event management," in *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7531 LNCS, pp. 171–180*, 2012.

[147] T. Li and J. Horkoff, "Dealing with security requirements for socio-technical systems: A holistic approach," in *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8484 LNCS, pp. 285–300*, 2014.

[148] A. Souag, C. Salinesi, R. Mazo and I. Comyn-Wattiau, "A security ontology for security requirements elicitation," in *International symposium on engineering secure software and systems (pp. 157-177). Springer, Cham.*, 2015.

[149] C. Neureiter, G. Eibl, D. Engel, S. Schlegel and M. Uslar, "A concept for engineering smart grid security requirements based on SGAM models," in *Computer Science-Research and Development, 31(1-2), 65-71.*, 2016.

[150] N. S. Rosa, G. R. Justo and P. R. Cunha, "A framework for building non-functional software architectures," in *Proceedings of the 2001 ACM symposium on Applied computing (pp. 141-147). ACM.*, 2001.

[151] J. Jürjens, "Using UMLsec and goal trees for secure systems development," in *Proceedings of the 2002 ACM symposium on Applied computing (pp. 1026-1030). ACM.*, 2002.

[152] D. Basin, J. Doser and T. Lodderstedt, "Model driven security for process-oriented systems," in *Proceedings of the eighth ACM symposium on Access control models and technologies (pp. 100-109). ACM.*, 2003.

[153] R. De Landtsheer and A. Van Lamsweerde, "Reasoning about confidentiality at requirements engineering time," in *Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering (pp. 41-49). ACM.*, 2005.

[154] J. Romero-Mariona, "Secure and usable requirements engineering," in *Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering (pp. 703-706). IEEE Computer Society.*, 2009.

[155] J. S. Cui and D. Zhang, "The research and application of security requirements analysis methodology of information systems," in *Anti-counterfeiting, Security and Identification, 2008. ASID 2008. 2nd International Conference on (pp. 30-36). IEEE.*, 2008.

[156] G. Howard, M. Butler, J. Colley and V. Sassone, "Formal Analysis of Safety and Security Requirements of Critical Systems Supported by an Extended STPA Methodology," in *Security and Privacy Workshops (EuroS&PW), 2017 IEEE European Symposium on (pp. 174-180). IEEE.*, 2017.

[157] S. Lipner, "The trustworthy computing security development lifecycle," in *Computer Security Applications Conference, 2004. 20th Annual (pp. 2-13). IEEE.*, 2004.

[158] N. Mead, E. Hough and T. Stehney, "Security Quality Requirements Engineering (SQUARE) Methodology," in *TECHNICAL REPORT CMU/SEI-2005-TR-009* , 2005.

[159] R. Fredriksen, M. Kristiansen, B. A. Gran, K. Stølen, T. A. Opperud and T. Dimitrakos, "The CORAS framework for a model-based risk management process," in *International Conference on Computer Safety, Reliability, and Security (pp. 94-105). Springer, Berlin, Heidelberg.*, 2002.

[160] J. Viega, "Building security requirements with CLASP," in *In ACM SIGSOFT Software Engineering Notes (Vol. 30, No. 4, pp. 1-7). ACM*, 2005.

[161] S. Rehman and V. Gruhn, "Security Requirements Engineering (SRE) Framework for Cyber-Physical Systems (CPS): SRE for CPS," in *New Trends in Intelligent Software Methodologies, Tools and Techniques: Proceedings of the 16th International Co*, 2017.

[162] P. Salini and S. Kanmani, "A Survey on Security Requirements Engineering," in *International Journal of Reviews in Computing, 8(1), 1-10.*, 2011.

[163] N. R. Mead, "Identifying security requirements using the security quality requirements engineering (SQUARE) method," in *Integrating Security and Software Engineering: Advances and Future Visions (pp. 43-69). IGI Global*, 2007.

[164] N. R. Mead, V. Viswanathan, D. Padmanabhan and A. Raveendran, "Incorporating security quality requirements engineering (SQUARE) into standard life-cycle models (No. CMU/SEI-2008-TN-006)," *CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.*, 2008.

[165] D. Munante, V. Chiprianov, L. Gallon and P. Aniorté, "A review of security requirements engineering methods with respect to risk analysis and model-driven engineering," in *International Conference on Availability, Reliability, and Security (pp. 79-93). Springer, Cham.*, 2014.

[166] L. Futcher and R. von Solms, "Guidelines for secure software development," in *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology (pp. 56-65). ACM*, 2009.

[167] T. Mir, A. Revuru, D. Manohar and V. Batta, "Threat analysis and modeling during a software development lifecycle of a software application," in *United States patent US 8,091,065.*, 2012.

[168] M. Howard and S. Lipner, The security development lifecycle, Redmond: Microsoft Press, 2006.

[169] S. H. Houmb, S. Islam, E. Knauss, J. Jürjens and K. Schneider, "Eliciting security requirements and tracing them to design: an integration of Common Criteria, heuristics, and UMLsec.," in *Requirements Engineering, 15(1), 63-93*, 2010.

[170] G. Popp, J. Jurjens, G. Wimmel and R. Breu, "Security-critical system development with extended use cases," in *Tenth Asia-Pacific Software Engineering Conference, 2003. (pp. 478-487). IEEE.*, 2003.

[171] J. P. Walton, "Developing an enterprise information security policy," in *Proceedings of the 30th annual ACM SIGUCCS conference on User services (pp. 153-156). ACM.*, 2002.

[172] H. Mouratidis and P. Giorgini, "Secure tropos: a security-oriented extension of the tropos methodology," in *International Journal of Software Engineering and Knowledge Engineering, 17(02), 285-309*, 2007.

[173] A. Belapurkar, A. Chakrabarti, H. Ponnapalli, N. Varadarajan, S. Padmanabhuni and S. Sundarrajan, "Distributed systems security: issues, processes and solutions," in *John Wiley & Sons*, 2009.

[174] A. Alkussayer and W. H. Allen, "The ISDF framework: towards secure software development," in *Journal of Information Processing Systems, 6(1), 91-106.*, 2010.

[175] K. Buyens, R. Scandariato and W. Joosen, "Process activities supporting security principles," in *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International (Vol. 2, pp. 281-292). IEEE.*, 2007.

[176] D. Mellado, E. Fernández-Medina and M. Piattini, " A common criteria based security requirements engineering process for the development of secure information systems.," in *In Computer Standards & Interfaces 29 (2), pp. 244–253. DOI: 10.1016/j.csi.2006.04.002.*, 2007.

[177] K. Labunets, F. Massacci and F. Paci, "An experimental comparison of two risk-based security methods," in *In Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on (pp. 163-172). IEEE*, 2013.

[178] A. Vorster and L. E. S. Labuschagne, "A framework for comparing different information security risk analysis methodologies," in *Proceedings of the 2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries (pp. 95-103).* , 2005.

[179] F. Den Braber, I. Hogganvik, M. S. Lund, K. Stølen and F. Vraalsen, "Model-based security analysis in seven steps—a guided tour to the CORAS method," in *BT Technology Journal, 25(1), 101-117.*, 2007.

[180] S. Rehman and V. Gruhn, "An effective security requirements engineering framework for cyber-physical systems," in *Technologies, 6(3), 65.*, 2018.

[181] S. Rehman, C. Allgaier and V. Gruhn, "Security Requirements Engineering: A Framework for Cyber-Physical Systems.," in *International Conference on Frontiers of Information Technology (FIT) (pp. 315-320). IEEE.*, 2018.

[182] D. Firesmith, "Specifying reusable security requirements," in *Journal of Object Technology, 3(1), 61-75.*, 2004.

[183] G. Sindre and A. Opdahl, "Eliciting security requirements with misuse cases," in *Requirements engineering, 10(1), pp.34-44.*, 2005.

[184] K. Pohl, Requirements engineering: fundamentals, principles, and techniques, Springer Publishing Company, Incorporated., 2010.

[185] C. Haley, R. Laney, J. Moffett and B. Nuseibeh, "Security requirements engineering: A framework for representation and analysis," *IEEE Transactions on Software Engineering,*, Vols. 34(1),, pp. pp.133-153., 2008.

[186] L. Wang, M. Törngren and M. Onori, "Current status and advancement of cyber-physical systems in manufacturing," in *Journal of Manufacturing Systems, 37, 517-527.*, 2015.

[187] V. Jirkovský, M. Obitko and V. Mařík, "Understanding data heterogeneity in the context of cyber-physical systems integration," in *IEEE Transactions on Industrial Informatics, 13(2), 660-667.*, 2016.

[188] H. Suleiman and D. Svetinovic, "Evaluating the effectiveness of the security quality requirements engineering (SQUARE) method: a case study using smart grid advanced metering infrastructure., 18(3), 251-279.," in *Requirements Engineering*, 2013.

[189] A. Souag, R. Mazo, C. Salinesi and I. Comyn-Wattiau, "Reusable knowledge in security requirements engineering: a systematic mapping study," in *Requirements Engineering, 21(2), 251-283.*, 2016.

[190] R. Baheti and H. Gill, "Cyber-physical systems," in *The impact of control technology, 12(1), pp.161-166.*, 2011.

[191] G. Disterer, "ISO/IEC 27000, 27001 and 27002 for information security management," 2013.

[192] E. Humphreys, "Implementing the ISO/IEC 27001 information security management system standard," in *Artech House, Inc..*, 2007.

[193] S. Rehman and V. Gruhn, "An approach to secure smart homes in cyber-physical systems/internet-of-things," in *Fifth International Conference on Software Defined Systems (SDS) (pp. 126-129). IEEE.*, 2018.

[194] Y. Wang, G. Attebury and B. Ramamurthy, "A survey of security issues in wireless sensor networks," 2006.

[195] R. Gandhi, A. Sharma, W. Mahoney, W. Sousan, Q. Zhu and P. Laplante, "Dimensions of Cyber-Attacks: Cultural, Social, Economic, and Political," in *IEEE Technol. Soc. Mag., vol. 30, no. 1, pp. 28–38*, 2011.

[196] D. Kushner, "The Real Story of Stuxnet," in *IEEE Spectrum: Technology, Engineering, and Science News*, 2013.

[197] Q. I. Sarhana, "Security attacks and countermeasures for wireless sensor networks: Survey," in *International Journal of Current Engineering and Technology, 3(2), 628-635.*, 2013.

[198] "https://www.getcybersafe.gc.ca/cnt/rsks/cmmn-thrts-en.aspx," [Online].

[199] R. Anderson and S. Fuloria, "Security Economics and Critical National Infrastructure," in *Economics of Information Security and Privacy Eds. Boston, MA: Springer US, 2010, pp. 55–66*, 2010.

[200] R. Sandhu and P. Samarati, "Access control: principle and practice," in *IEEE Commun. Mag., vol. 32, no. 9, pp. 40–48*, , 1994.

[201] Y. Ren, V. A. Oleshchuk, F. Y. Li and X. Ge., "Security in mobile wireless sensor networks-A survey," 2011.

[202] Y. Xiao, Security in distributed, grid, mobile, and pervasive computing, CRC Press., 2007.

[203] Y. Yan, Y. Qian, H. Sharif and D. Tipper, "A survey on cyber security for smart grid communications," in *IEEE Communications Surveys & Tutorials, 14(4), 998-1010.*, 2012.

[204] I. Sommerville, "Software engineering 9th Edition," in *ISBN-10, 137035152.*, 2011.

[205] J. McGhee and M. Goraj, "Smart high voltage substation based on IEC 61850 process bus and IEEE 1588 time synchronization," in *First IEEE International Conference on Smart Grid Communications (pp. 489-494). IEEE.*, 2010.

[206] Z. Guo, N. Karimian, M. M. Tehranipoor and D. Forte, "Hardware security meets biometrics for the age of IoT," in *IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1318-1321). IEEE*, 2016.

[207] D. M. K. Jain, "Wireless Sensor Networks: Security Issues and Challenges," in *Int. J. Comput. Inf. Technol., p. 6*, 2011.

[208] L. Sha, S. Gopalakrishnan, X. Liu and Q. Wang, "Cyber-physical systems: A new frontier. In Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC'08," in *IEEE International Conference on (pp. 1-9). IEEE*, 2008.

[209] R. Shorey, A. Ananda, M. C. Chan and W. T. Ooi, "Mobile, wireless, and sensor networks: technology, applications, and future directions," in *John Wiley & Sons.*, 2006.

[210] J. S. Lee, Y. W. Su and C. C. Shen, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *Industrial electronics society, 5, 46-51.*, 2007.

[211] K. Pothuganti and A. Chitneni, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *Advance in Electronic and Electric Engineering, 4(6), 655-662.*, 2014.

[212] G. Stoneburner, A. Y. Goguen and A. Feringa, "Risk management guide for information technology systems," in *Special publication 800-30, National Institute of Standards and Technology*, 2002.

[213] Caralli, Richard, J. F. Stevens, L. R. Young and W. R. Wilson., " Introducing octave allegro: Improving the information security risk assessment process," 2007.

[214] D. J. Landoll and D. Landoll, "The security risk assessment handbook: A complete guide for performing security risk assessments," in *CRC Press.*, 2005.

[215] R. Schmidt, K. Lyytinen, M. Keil and P. Cule, "Identifying software project risks: An international Delphi study," in *Journal of management information systems, 17(4), 5-36.*, 2001.

[216] L. Xiaosong, L. Shushi, C. Wenjun and F. Songjiang, "The application of risk matrix to software project risk management," in *International Forum on Information Technology and Applications (Vol. 2, pp. 480-483). IEEE.*, 2009.

[217] E. R. Stroie and A. C. Rusu, "Security risk management-approaches and methodology," in *Informatica Economica, 15(1), 228.*, 2011.

[218] J. Rumbaugh, "Getting started: using use cases to capture requirements," in *Journal of Object Oriented Programming 7(5):8–23*, 1994.

[219] H. El-Hadary and S. El-Kassas, "Capturing security requirements for software systems.," in *Journal of advanced research, 5(4), 463-472.*, 2014.

[220] K. Weidenhaupt, K. Pohl, M. Jarke and P. Haumer, "Scenarios in system development: current practice," in *IEEE software, 15(2), 34-45.*, 1998.

[221] I. Jacobson, Object-Oriented Software Engineering: A Use Case Driven Approach, Pearson Education India., 1992.

[222] L. L. Constantine, L. A. Lockwood and L. Wood, Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design, SIGCHI Bulletin, 32(1), 111., 2000.

[223] A. Cockburn, Writing effective use cases, The crystal collection for software professionals, Cockburn, A. (2000). Writing effective use cases: Addison-Wesley Professional Reading, 2000.

[224] D. Kulak and E. Guiney, Use Cases: Requirements in Context, Addison-Wesley, 2012.

[225] G. Sindre and A. L. Opdahl, "Capturing security requirements through misuse cases," in *NIK 2001, Norsk Informatikkonferanse 2001, http://www. nik. no*, 2001.

[226] "https://soccerwatch.tv/aboutus.," [Online].