Computers & Security 124 (2023) 102950

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Threat Modeling of Cyber-Physical Systems - A Case Study of a Microgrid System

Shaymaa Mamdouh Khalil[a,*], Hayretdin Bahsi[a], Henry Ochieng' Dola[a], Tarmo Korõtko[b], Kieran McLaughlin[c], Vahur Kotkas[a]

[a] School of Information Technologies, Tallinn University of Technology, Estonia
[b] FinEst Centre for Smart Cities, Tallinn University of Technology, Estonia
[c] Centre for Secure Information Technologies, Queen's University Belfast, UK

ARTICLE INFO

Article history:
Received 6 July 2022
Revised 31 August 2022
Accepted 7 October 2022
Available online 12 October 2022

Keywords:
Threat modeling
Cyber-Physical System (CPS)
Industrial Control Systems (ICS)
STRIDE
Microgrid
Impact assessment

ABSTRACT

Cyber threat modeling is an analytical process that is used for identifying the potential threats against a system and supporting the selection of security requirements in the early stages of the system development life cycle. Thus, threat modeling is a vital instrument for the realization of the secure-by-design principle. Despite being a well-known practice in software development projects, its adaptation to cyber-physical systems still requires systematic elaboration. The complex interactions between cyber and physical spaces and their reflection on the cyber threat landscape constitute a significant challenge for the system development teams. This study proposes a detailed methodology to apply STRIDE to cyber-physical systems and demonstrates its applicability in a case study of a microgrid system. Our methodology provides a systematic threat elicitation procedure based on an attack taxonomy that was created for this research. This paper also shows how assets could be identified, data flow diagrams formed, trust boundaries determined, and threats prioritized, in the case of a cyber-physical system.

1. Introduction

Cyber-Physical Systems (CPSs) play an increasing role in various critical infrastructures (e.g., energy, transportation, health, etc.) due to the significant developments in computing and communication fields. Although much attention has been paid to the safety aspect of these systems during the system development and validation efforts, the malicious acts of human beings in terms of cyber threats are often neglected. Various real-world examples and research studies demonstrated that cyber-attacks constitute a real threat against these systems and may induce potential harm including significant physical consequences (Canaan et al., 2020), while systematically identifying threats across the cyber and physical domains remains a challenge.

The application of security-by-design principles during the development of these complex systems is essential to have an acceptable level of security assurance (ISAGCA, 2020). Threat modeling is one of the core tasks that aim to analyze the applicable threats to elicit security requirements at the early stages of the development life-cycle. Despite being a well-known practice in IT systems, threat modeling has not been widely adapted to the life-cycle of CPSs. One of the main reasons is that although the existing threat modeling methods guide well in identifying the information-centric results in software-based systems, they are not well-equipped for exploring the interaction between cyber and physical spaces within the framework of CPSs (Fernandez, 2016; Jamil et al., 2021).

It is important to note that the threat modeling endeavor is not purely analytical. It is also a process that facilitates communication and common understanding between different stakeholders. When compared to software-based systems, cyber-physical systems have more varied hardware, software, and communication components fulfilling a physical task, thus, necessitating the involvement of more divergent stakeholder groups such as operators dealing with the physical processes or more heterogeneous development teams with various backgrounds besides the IT or OT system operators. Therefore, threat modeling methods, which have been mostly developed for software, should be adapted to the stakeholder structure of the CPS environments.

In a typical software-based system, the identified assets are directly reflected in the system model and usually the main question left is how to draw the trust boundaries. On the other side, in a CPS, some assets may not have any computing component,

* Corresponding author.
E-mail address: shaymaa.khalil@taltech.ee (S.M. Khalil).

https://doi.org/10.1016/j.cose.2022.102950

which may disqualify them from the studies solely focusing on cyber threats. A clear definition is required for how a cyber component would be differentiated from a physical one. However, the demonstration of physical components and their interactions with the cyber ones may still be needed in system models, as comprehending the impact of cyber threats would be easier for various experts if they had the whole picture. Note that the experts dealing with physical processes are significant contributors to threat modeling, especially during the assessment of threat consequences. Thus, system modeling should consider all the stakeholders. This aspect has not been considered in detail by the studies in the literature.

The asset identification is one of the initial stages of threat modeling. The determination of information assets is highly significant as they play a critical role in comprehending threat consequences and spotting information flows. Although practitioners have a good understanding of the definition of information assets in a software-based system, it is not as clear how to perceive and categorize them in a CPS.

The threat models of software-based systems usually disregard physical access or include strong assumptions about the protection of physical security. Contrarily, a threat actor can physically access the components of a CPSand compromise the device by using different hardware interfaces or accessing the network channels, as the whole system may be located in different places without proper physical security countermeasures. Therefore, in addition to logical controls, physical controls are significant determinants in the identification of trust boundaries. The literature do not thoroughly show how the boundaries are decided in the light of logical and physical security assumptions in CPSs. More importantly, they do not clarify the assumptions about the environment and the addressed system, and link such assumptions with the trust boundary considerations. This is especially important for conducting threat modeling studies within well-accepted frameworks and standards. Moreover, the interaction between the cyber and physical spheres requires more careful consideration as some of the cyber-attacks may have some preconditions related to physical security.

Threat modeling methods (e.g., STRIDE) do not have strong threat enumeration constructs beyond their system models. Thus, the success of the threat elicitation usually depends on the experience of the experts involved in the work (Jamil et al., 2021; Scandariato et al., 2015). However, a knowledge base in the form of threat databases, attack lists/taxonomies, or vulnerability databases is an essential construct that increases the outcome quality of the threat enumeration and partly eliminates the dependence on highly-experienced experts. Security teams can convert the experience they gathered throughout various projects into such a knowledge base, leading to a continuous improvement in their processes. Additionally, this construct would be integral to any semi- or fully-automatized threat modeling efforts. The studies in this domain do not demonstrate how a knowledge base could be integrated into the threat elicitation stage.

The threat modeling studies do not usually explain how their outcome would be used in the secure development life-cycle. Although the main intention is to determine security requirements in the early stages, they do not have clarity on whether they generate security requirements or security objectives, do not thoroughly describe the security context (i.e., environment, assumptions, complementary controls), and do not show how the results would be integrated to the frameworks of standards (e.g., Security for Industrial Automation and Control Systems IEC 62443). As expected, these general-purpose standards do not propose a specific threat modeling method. However, it is important to showcase to the practitioners how the results of the threat modeling process can be linked to the standards.

We conclude that there is a gap in CPSs threat modeling case studies that address the interaction between cyber and physical spaces, with a clear and repeatable threat modeling methodology. This research aims to answer two main research questions:

RQ(1) How can we create a systematic CPSsthreat modeling methodology that could be applied by security experts and practitioners in the real world?

RQ(1.a) What are the main stages of such a systematic CPSs threat modeling methodology?

RQ(2) How could we enhance some of the currently used techniques in threat modeling (e.g. Data Flow Diagram (DFD)) to make them more relevant to CPSs?

To answer these research questions, we propose a comprehensive threat modeling methodology that adapts and extends STRIDE for CPSs and demonstrates the results of its application to a microgrid system, which is selected as a case study. Thekey contribution points of this study could be summarized as follows:

(1) Presenting a systematic and detailed threat modeling methodology to apply STRIDE in CPSs, and supporting it with a real-world case study.

(2) Providing detailed DFD that shows the interaction between the cyber and physical spaces. While proposing procedures for asset identification and trust boundaries determination.

(3) Establishing a procedure that incorporates an attack taxonomy into the threat elicitation phase for more systematic threat identification. We complement this procedure with a task that maps the threats to the potential consequences including the ones in the physical space.

(4) Proposing a threat prioritization procedure to identify critical and high-priority threats.

(5) Showing how the threat modeling results can be used for identifying the security requirements within the framework of the IEC 62443 standard.

The rest of this paper is organized as follows: Section 2 discusses related work, Section 3 provides an overview of the proposed threat modeling methodology, Section 4 provides an introduction to the case study, Section 5 presents the results of the nine stages of the case study, Section 6 discusses the proposed methodology and its limitations, and Section 7 concludes the paper.

2. Related work

Threat modeling has various definitions, while a relevant definition of threat modeling is: *A process that can be used to analyze potential attacks or threats, and can also be supported by threat libraries or attack taxonomies* (Xiong and Lagerström, 2019). The literature proposes various CPSs threat modeling methodologies that serve different requirements and approaches of researchers and practitioners. The choice of a suitable threat modeling method depends on the stakeholders' experience, objectives, time, and tools. While there are many methods proposed in the literature, few methods are used in practice (e.g., STRIDE, LINDDUN). The study (Jamil et al., 2021) shows that when performing CPSs threat modeling, some practitioners use known methods then elaborate on them based on their experience. As per the same study, practitioners also combine known methods and approaches, and they sometimes use their own risk assessment techniques in CPSs threat modeling, which indicates a need for creating a more structured methodology that could be widely used in organizations.

STRIDE is a mnemonic that was first introduced by Microsoft for threat enumeration and it stands for **S**poofing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of service, and **E**levation of privilege. Modeling threats using STRIDE requires a representation of the system under consideration, predominantly realized through the use of a data flow diagram. STRIDE analyses the system components against 6 main security properties

(confidentiality, integrity, availability, authentication, authorization, and non-repudiation). However, the literature shows that there is no clear standard procedure to apply STRIDE on a CPS.

The study (Khan et al., 2017) provides a STRIDE-based use case related to the smart grid domain. The study proposed a five high-level steps methodology to apply STRIDE on a CPS, which consists of: (1) Decompose the system into components, (2) Plot DFD for system components, (3) Analyze threats in the DFD, (4) Identify vulnerabilities, (5) Plan mitigation strategies. While Khan et al. (2017) provides a lightweight approach for STRIDE application on CPSs, the study elicits threats based on a list of threat consequences that are identified by experts at the beginning of the stage *analyze threats in DFD*. The proposed procedure for threat analysis seems to be challenging when applied by a team that does not have experienced persons in both electrical and security fields, due to the identification of threat consequences before threat elicitation.

Several studies complement STRIDE with risk assessment techniques to prioritize the elicited CPSs threats. For example, the authors of (Haider et al., 2019) applied STRIDE to five common types of attacks that can affect Advanced Metering Infrastructure (AMI), and used DREAD for threat prioritization. The study (Ramis Ferrer et al., 2017) presents a STRIDE case study based on a proposed solution for controlling industrial processes. The research prioritizes the threats based on a risk assessment model, where the risk is calculated by the multiplication of *Likelihood* and *Impact*, and a value on a scale of (0–10) is assigned for both *Likelihood* and *Impact*. While, the study (Ramis Ferrer et al., 2017) does not clarify how the likelihood values were decided. On the other hand, Ahn et al. (2021) applies STRIDE on a Power Transformer Diagnosis System (PTDS), and proposes a STRIDE threat model chart, where each STRIDE threat is mapped to a *Common Vulnerability Enumeration* (CVE) and *Common Vulnerability Scoring System* (CVSS) of a specific vulnerability, which could be used in threats prioritization. However, the fidelity of using specific vulnerabilities scores similar to CVSS scores in the early stages of the CPS life-cycle is debatable and might provide misleading results, as CVSS vulnerabilities are vendor and version specificand such information is not available in the early stage of a development project.

STRIDE mainly focuses on the security properties of the system (i.e., confidentiality, integrity, availability, authentication, authorization, non-repudiation). However, there exist some methods that focus on other aspects such as privacy or safety. For example, the LINDDUN threat modeling method supports the identification of privacy weaknesses in systems under development. The paper (Wuyts et al., 2014) provides an empirical validation study for LINDDUN, where the method was applied to a smart-grid system. On the other hand, some methods focus on the safety of the system under development (e.g., STPA), and they are extended by researchers to cover both the safety and security aspects of a CPS. For example, the study (de Souza et al., 2020) extends STPA with STRIDE. Also, Friedberg et al. (2017) propose STPA-SafeSec threat modeling method that covers both safety and security aspects of a CPS, and presents the method with a use case related to the smart grid domain.

As presented in this section, the use of STRIDE in threat modeling of CPSs has been proposed in several academic papers, although the procedure for its application is not well documented, thus, its implementation details vary in the literature. When comparing STRIDE to LINDDUN (Wuyts et al., 2010), we find that LINDDUN provides detailed procedures on how to apply the method, and it also provides a threat tree catalog, which provides attack paths for a specific LINDDUN threat category[1]. Detailed documentation, including a tutorial for LINDDUN (Wuyts and Joosen, 2015), makes it easy to run LINDDUN for practitioners that are not familiar with the method, which is not the case for STRIDE.

The CPS use cases proposed in the literature tend to be theoretical or applied on testbeds, while there is a clear gap in papers presenting real-world case studies. The literature shows that there is a lack of detailed and repeatable threat modeling methodologies that highlight the procedures related to the trust boundaries identification, threat modeling assumptions, as well as threat elicitation and prioritization.

3. Method

To address the identified research gaps, we propose a nine-stage CPSs threat modeling methodology, as shown in Fig. 1, and we validated the proposed methodology by a real-world case study of a microgrid system. The first stage of the proposed methodology, *Initial Attack Taxonomy Creation*, consists of reviewing the literature regarding the attacks on similar systems (e.g., microgrids and smart grids), and creating an initial attack taxonomy (see Section 5.1 for the case study results of the attack taxonomy stage). This step assists the cyber security experts to get familiar with the targeted environment (e.g., microgrid) and its related security issues. In future runs of the proposed methodology, the first stage could be reduced or eliminated, depending on the CPS application area. We propose an attack taxonomy that could be used as a baseline for microgrid and smart grid threat modeling, while it might require some updates in case of a different CPS application area (e.g. autonomous vehicles).

The main goal of the second stage, *Information and System Assets Identification*, is the identification of all information and system assets, regardless of whether they would be in the scope of the threat modeling exercise or not. At this stage, although it is required to identify and understand the system assets in detail, we put a specific emphasis on listing and categorizing the information assets. When compared to a software-based system, a cyber-physical one has some significant nuances. Taking a microgrid system as an example, we find that such a system mainly operates

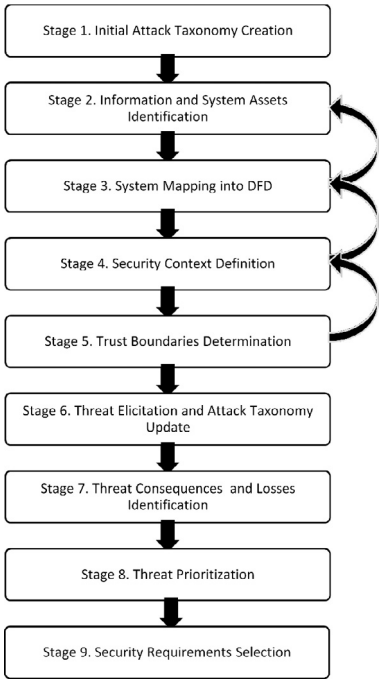

Fig. 1. Main Stages of the Proposed CPS Threat Modeling Methodology.

[1] https://www.linddun.org/linddun.

on two distinct information categories, control and measurement, which may have varied security priorities depending on the controlled physical process and timing requirements (e.g., real-time vs non real-time). Additionally, advanced analytical capabilities integrated into a microgrid system require additional information categories such as information from external resources (e.g., weather data, energy price information), or analysis outputs regarding the prediction of energy usage or production, as detailed in the case study results, Section 5.2. The goal of this stage could be achieved through an initial system architecture diagram, where system assets could be identified first. Then the threat modeling team could discuss the potential information assets that would reside in the system.

The data flow diagram is created in the third stage, *System Mapping into DFD*. It is important to note that, although the second and third stages are shown as distinct stages in Fig. 1, there might be many feedback iterations between these two stages during the team discussions. Both stages could even be considered as one integrated stage. However, instead, we showed them as two for the sake of clarity. Another important note regarding the system mapping stage is that all experts including the system architects should actively take part in discussions. Tracking the life-cycle of the information assets and the implications on the information flow constitute the main discussion points in that collaborative effort. On the other side, it is important to determine the physical system components which do not have any computational capability, and the analog information flows. Although the threat modeling scope is to identify the cyber-related threats, we propose to show physical assets and analog flows on the DFD as they could be informative to the specialized system experts (e.g., power systems experts in the case of a microgrid) when they give feedback whileidentifyingthe threat consequences (Stage 7). The case study results of the third stage are given in Section 5.3.

The fourth stage, *Security Context Definition*, is launched after reaching a level of maturity in the DFD. At this stage, the threat modeling team agrees on the key physical security assumptions, the trusted entities (e.g., system admins), identify the main threat actors that might target the system (e.g., skilled cyber criminals), and finally agree on the excluded attacks (e.g., supply chain attacks).

The outcomes of the fourth stage play a key role in the fifth stage, *Trust Boundaries Determination*. We propose good documentation for the security context identified in Stage 4, followed by a documented reasoning for the choice of trust boundaries. Such documentation would ease the re-consideration of boundary decisions, in case of system modifications or new security context identification. Note that, similar to the second and third stages, the fourth and fifth stages can be also considered as integrated stages as there might be various feedback loops between them. In some cases, there might be a need to return to the second and third stages from the fourth and fifth stages to tweak asset lists and DFD. All of the team members should cooperate in the discussions of these stages and agree on the results. Sections 5.4 and 5.5 give detailed discussions about the realization of Stage 4 and Stage 5 in our case study.

Table 1
Applicable Threats to DFD Elements (Khan et al., 2017; Shostack, 2008).

DFD Element	S	T	R	I	D	E
Entity	✓		✓			
Data Flow		✓		✓	✓	
Data Store		✓	✓	✓	✓	
Process	✓	✓	✓	✓	✓	✓

The sixth stage, *Threat Elicitation and Attack Taxomy Update*, is mainly devoted to the elicitation of the threats based on the STRIDE-per-element approach in which each component in the DFDs is analyzed individually. An alternative method, STRIDE-per-interaction, focuses on the information flows crossing the trust boundaries. However, this STRIDE variant is more complex than the former variant. STRIDE-per-element is considered as a clear and easy STRIDE variant to adapt, especially for individuals with limited knowledge about threat modeling (Shostack, 2014). The results of the experiment (Tuma and Scandariato, 2018) show that STRIDE-per-element has a higher level of completeness when compared to STRIDE-per-interaction. In critical infrastructure, especially power systems, eliciting the highest possible number of threats would give the decision owners better visibility about the possible threats and their impact on the system, then they can decide either to accept such threats or apply a countermeasure for better defense of the system. For these reasons, we promote the use of STRIDE-per-element, especially for systems with a moderate size of assets, as passing over each component could be feasible in a reasonable time. A proposed solution to optimize the use of STRIDE-per-element on CPSs, is eliminating the analysis of data flows where both endpoints are in the same trust zone, which should decrease the number of elicited threats. Such an approach requires the analysis of every process and data store within the same trust boundary, to assess the impact of the threats on each of these components. STRIDE-per-interaction focuses on the information flows; thus, it may not identify the threats emanating from physical or cyber access to individual devices. Using the proposed STRIDE-per-element optimization approach would enable the threat modeling team to consider the situations of each device regarding physical access and its cyber consequences. Table 1 could be used for mapping the DFD elements to the threat categories of STRIDE, when using STRIDE-per-element approach.

We propose a threat elicitation procedure that incorporates the attack taxonomy and information assets into the analysis as given in Fig. 2. We utilize an elicitation order in which a DFD element should be first selected, the practitioners should identify the relevant information assets, pick the threat category and then check the attack taxonomy for the applicable attack types. This

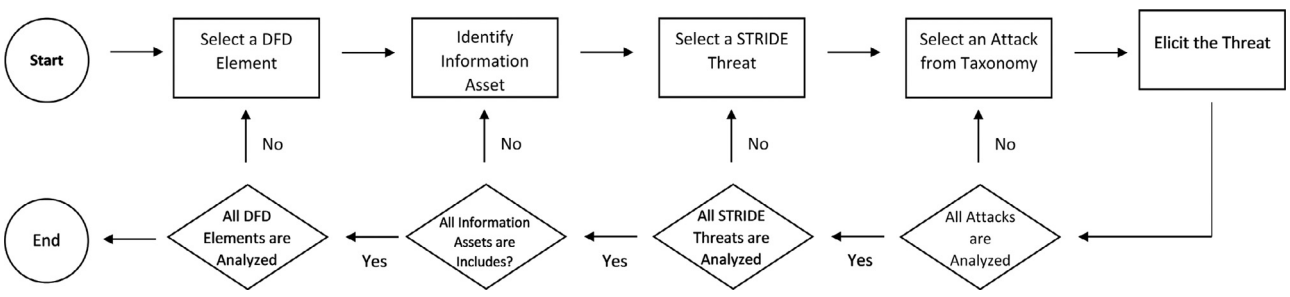

Fig. 2. Threat Elicitation Procedures.

systematic elicitation procedure enables the creation of default threat templates that can be useful for the experts and the ones who aim to create more automatized threat modeling tools. Details of the application of the threat elicitation procedures are given in Section 5.6. The discussions about threat elicitation are mostly done among cyber security experts. However, intermediate and final results should be shared with the other experts for obtaining further opinions. When compared to the previous stages (e.g., asset identification, and system mapping), the contribution of the system owner and system architects is relatively limited. This stage is concluded by a review and update of the attack taxonomy.

The seventh stage, *Threat Consequences and Losses Identification* is mostly done by the system experts with the guidance of the cyber-security experts, as discussed in the case study Section 5.7. The team should go through each of the elicited threats and identify its threat consequences, as well as relatedpotential tangible and intangible losses.

The results of the threat modeling process should provide the system developers and other stakeholders with a list of security requirements, which could be applied to the system to enhance its security. To reach this point, it is useful to prioritize the threats based on their consequences and potential losses, as such information could support the stakeholder in the security requirements choice. *Threat Prioritization* and *Security Requirements Selection* consist the stages eight and nine of the proposed threat modeling methodology, as detailed in the case study Sections 5.8, and 5.9.

4. Introduction to the microgrid case study

We conducted this study at the conceptual analysis and requirement elicitation stages of a project that aims to develop a software platform for effective energy management in a microgrid system and evaluate it in an industrial site. This project also covers a system integration effort that requires the procurement and installation of various cyber-physical components. The main objective of the threat modeling in this study is to identify the low-level security objectives that lead to security requirements for the system components and software platform.

We composed a threat modeling team of five experts, i.e., three cyber security experts, one software architect, and one expert in power systems. The power systems expert was working closely with the industrial system owner and was also responsible for coordinating efforts about deriving all functional requirements and conducting the procurement preparations. Therefore, in our context, we assumed that he had two main roles, power systems expert and system owner. Note that he is referred to as system owner or power system expert, interchangeably, in the remaining part of the paper.

The system owner provided a rough initial system architecture in the form of a block diagram, as given in Fig. 3, and shared it with the rest of the team for initiating the asset identification discussion. This diagram mainly included some key system assets, the electrical connections (represented by black lines in Fig. 3), and possible network segments for control and measurement data communication represented in different colors. Then the security team formed an initial list of system and information assets.

The hardware structure of the studied microgrid includes a microgrid control unit (mapped as *SERVER* in Fig. 3), which consists of computation equipment and the software platform and is responsible for data and log management, as well as other services (e.g., monitoring). The system also includes an Energy Storage System (ESS), which is used for storing excess energy and supplying it when required. The ESS can be used to maximize the use of locally produced energy and/or provide capacity management, but also to provide ancillary services like the improvement of microgrid power quality, voltage support, and restoration. The Moulded Case Circuit Breaker (MCCB), Intelligent Electronic Device (IED), and Remote Terminal Unit (RTU) are power equipment required to connect to the Distribution System Operator (DSO) grid through the Point of Common Coupling (PCC). The system also includes PhotoVoltaic (PV) generation and might facilitate one or more prosumers, which are proactive and intelligent entities that either produce or consume energy (Korõtko et al., 2019). The metering of energy consumption is carried out through energy meters, while the measurement and operational monitoring of different power system properties are carried out using multimeters. Data concentrators are used for collecting and aggregating data

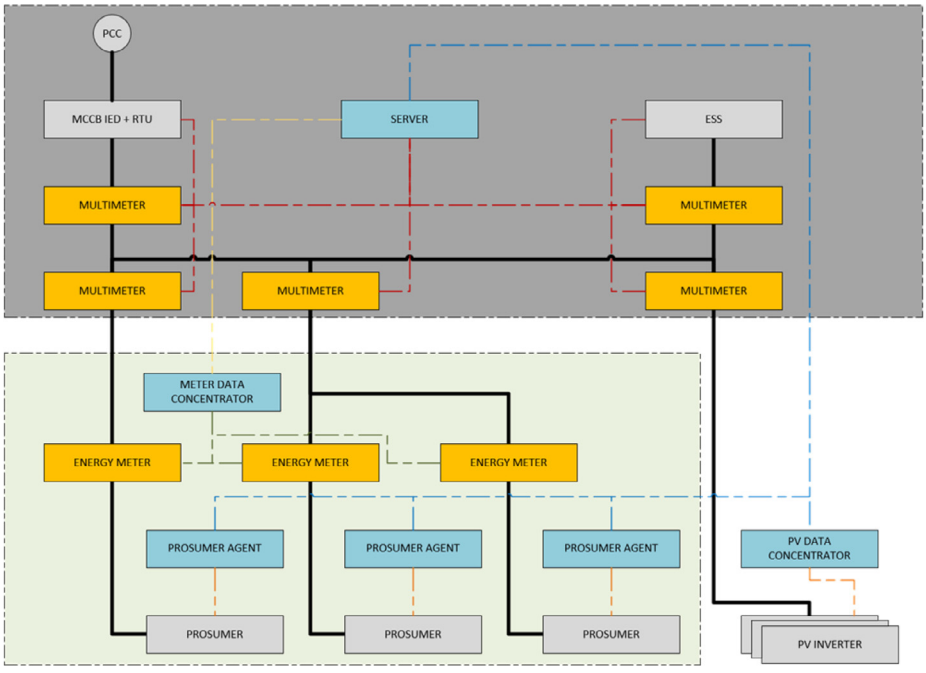

Fig. 3. Microgrid Initial Block Diagram.

from different parts of the system and communicating it with the server.

5. Case study results

5.1. Attack taxonomy

In our case study, we considered the attack taxonomy as a dynamic knowledge base to better systematize the threat elicitation procedure. During the whole study, we kept updating this knowledge base whenever we identified new sources in the literature and also reflected the results of discussions that occurred among the experts.

The attacks can be defined in different abstract levels. For instance, *ATT&CK for Industrial Control Systems*, which is a framework developed by MITRE for the classification and characterization of adversary behavior (Alexander et al., 2020), provides detailed attack tactics and techniques addressing industrial control systems. We consider that our attack categories should be more abstract as the details of the attack techniques do not contribute to our case study andthreat identification is conducted without knowing the system details.

The academic and grey literature have various documents listing the threats applicable to the whole system or specific components of smart grids. We mostly surveyed the academic literature while the initial attack taxonomy is published in Bahsi et al. (2022). After the elicitation of the threats (Stage 6), we utilized some studies that we did not use during the creation of the initial taxonomy for cross-checking our findings. Our aims were twofold: (1) The identification of the threats that we missed, and (2) The update of our attack taxonomy. The following documents were utilized for cross-checking the threat taxonomy: An academic study about the threat modeling of smart grid systems (Suleiman et al., 2015), protection profiles for Smart Metering, its gateway, and infrastructure (EN/CENELEC/ETSI, 2019; Muhammet Oztemur, 2014; SMGW-PP, 2014), and ENISA's security guideline (Egozcue et al., 2012).

In the final taxonomy, we decided to have two different *Integrity Attacks* types, *Forging* and *Tampering*, which were once only grouped in one attack type, *False Data Injection*. Moreover, we realized that, in the initial taxonomy, we considered the *Availability Attacks* only from the network point of view. We added two device-based availability attacks, *Delete/Corrupt Data* and *System Corruption*. The attacks *Credentials Compromise* and *Physical Compromise* were also added to the taxonomy.

The final attack taxonomy that we consolidated at the end of the study is given in Fig. 4. We classified the attacks into 2 groups. The first group, *Threat Precondition*, includes the *Device Compromise* and *Network Compromise* categories, and constitutes the precondition of a threat, meaning that the threat agent should get a foothold in the device or network before inducing the final harm to the assets. The second group, *Threat Final Harm*, includes the *Integrity* and *Availability Attacks* categories and defines how actual harm is done. Each threat definition covers mainly two attack types, one from the threat preconditions category and one from the threat final harm category.

We did not assign a specific category for confidentiality attacks as the attacker can easily access the data after compromising a device or network. For instance, credential compromise may easily lead the attacker to extract data from the device, or, the attacker accesses the data after conducting eavesdropping in the network.

Although we showed them in the taxonomy, we did not use some attack sub-categories in our case study due to the assumed properties of the target system. The system owner does not foresee a system component with wireless capabilities, thus, we did not use the *Jamming Attacks* which are defined as wireless attacks conducted at the physical level of the OSI Model. At this stage, we do not expect to have a dynamic network structure that may lead to availability attacks regarding the sub-category, *Routing Protocols Attacks*.

5.2. Assets identification

At the second stage of our threat modeling methodology, *Information and System Assets Identification*, we first identified the list of

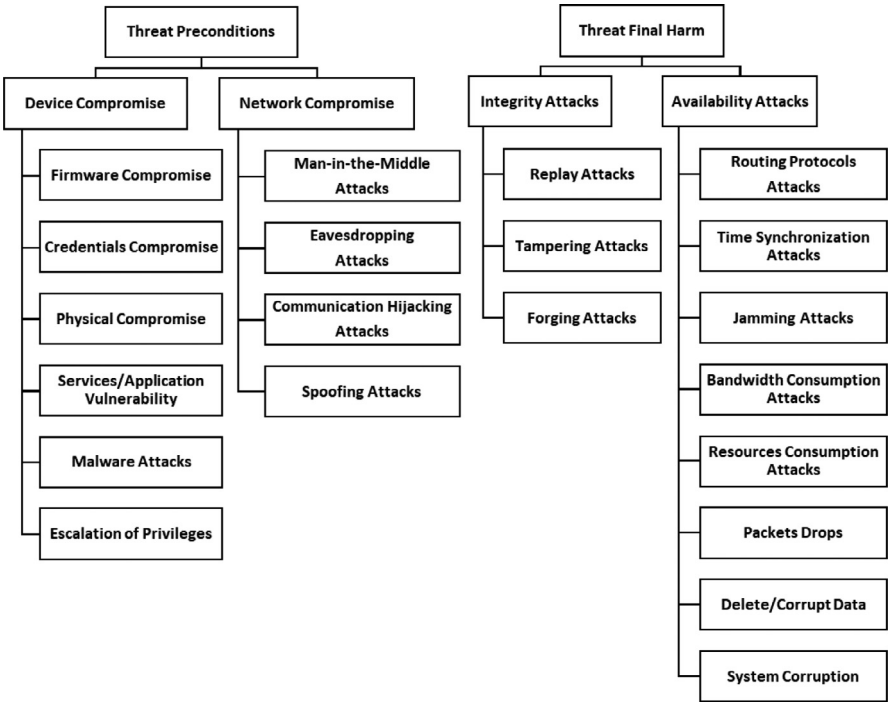

Fig. 4. Proposed Attack Taxonomy for Microgrid.

Table 2
List of Identified System Assets.

Asset Type	Asset ID	Asset Name
Software Related Process	P-01	Server
	P-02	Meter data concentrator
	P-03	PV data concentrator
	P-04	Prosumer gateway
	P-05	Prosumer Agent
Measurement Device Process	P-06	Prosumer multimeter
	P-07	Prosumer energy meter
	P-08	PV multimeter
	P-09	PCC multimeter
	P-10	ESS multimeter
Electrical Power Process	P-11	ESS control system
	P-12	MCCB control system
	P-13	PV inverter control system
	P-14	Prosumer control system
Data Store	S-01	Server data store
	S-04	Meter data concentrator data store
External Entity	E-01	Microgrid admin
	E-02	Microgrid admin computer
	E-05	DSO SCADA
	E-06	Prosumer admin
	E-07	External data sources
Physical Process	PH-01	Prosumer physical process
	PH-02	PV inverter physical process
	PH-03	ESS physical process
	PH-04	PCC physical process

system assets throughclose collaborations with system owners, discussions with developers, as well as the diagram shown in Fig. 3. As shown in Table 2, we categorized the system assets based on asset types, and for each asset, we have marked an asset ID that could be used to tag the asset. The System assets icludes two asset types that are usually used in a DFD, which are *External Entity* and *Data Store*. A DFD also contains a *Process* asset type, while we categorized the processes based on their functions, which resulted in four different types of CPSs processes: *Measurement Device Process, Electrical Power Process, Software Related Process*, and *Physical Process*.

Following the identification of system assets, we started to discuss the information assets and their categories. We identified thirteen distinct information asset types, categorized into five main categories, as shown in Table 3. We identified two time-sensitive information assets categories: *Control Data*, which is related to the control commands, and *Operational Data*, which contains also some time-sensitive information, as it provides visibility about the system and operation status. The operational data category includes three sub-categories, *Measurement Data, Alarm Data*, and *Event Data*. Another information asset category is the *Metering Data*, which is mainly used for commercial purposes as it provides the information required for billing. The fourth information asset category is the *Produced Data*, which includes two sub-categories of information assets: The *Prediction Data* which is processed data that is used to predict some values based on the acquired information, and the *External Data* (e.g. weather forecasts and electricity prices) that are collected from external data sources. The last information asset category is the *Device Management Data*, which further splits into three sub-categories: *Service Data* (i.e., admin access for maintenance purposes such as system update), *Access Logs*, and *Device Configuration*.

5.3. Data flow diagram (DFD)

Following the identification of system and information assets, we created what we named a *Data Flow Mapping Table* to map the data flow between the different system assets, as such information is essential for the DFD creation. Our initial idea was to create a separate data flow for each information asset type. While following the identification of 13 information asset types, and to avoid creating a complex DFD, we decided to assign a single data flow ID based on the corresponding system assets. However, we added a separate data flow for each communication direction, as shown in Table 4. Note that the table marks the direction of the data flow with the columns *From* and *To*, while we also marked in bold the system asset initiating the network connection, which might not be the same direction of data flow, as shown in the provided example entry. Such information might help the threat modeling team in eliminating irrelevant threats during the threat elicitation phase, as we would discuss further in this section.

By default, a DFD has four main elements (Shostack, 2008): *Process, Data Store, Data Flow*, and *External Entity*, as represented in Table 5. However, to make the DFD more comprehensive and relevant to CPS components, we propose some enhancements to the default DFD elements, which would be justified in the rest of this section. We created the microgrid system DFD using Lucidchart, where a detailed definition for each DFD element could also be found[2]. The DFD went through multiple iterations to reach the version shown in Fig. 5, as the assets lists were updated several times, based on continuous discussions with all the involved stakeholders.

The literature does not show clear procedures on whether physical system components should be included in a DFD, if so, how this should be done. Our microgrid system, which is also considered as an Industrial Control System (ICS), contains many physical processes that are managed by control systems, even though these do not contain any cyber component. We contemplate that DFD should still represent these processes and their analog interactions with cyber components although they may not directly induce cyber threats. In this way, it is easier for the practitioners, especially power system experts, to understand the dependencies between the cyber and physical components, which would help them to better assess the impact of each threat on the physical process.

To this end, we decided to represent physical components and their analog connections with other cyber components on the DFD. We demonstrated physical processes using a different color and represent them with a new DFD component type that was identified as *Physical Process*. We also differentiated the flows passing through digital and analog connections by using different arrow types, which gives the reader a better visual understanding of the system.

For some data flows, the system owner provided information about the direction of network connection initiation, which is not the same as the data flow direction. Such information was not normally mapped on a DFD. We decided to include this information in our DFD, to give the contributors a better understanding of the system as it would impact the elicited threats. For example, in our case, the server initiates the connections for all the system components, and no process can initiate the connection to the server, which eliminates some threats similar to targeting the resource consumption of the server through opening many concurrent connections from the same source.

Our DFD includes 41 data flows, where 26 are related to digital connections, while 15 are related to analog connections. Each data flow was tagged with a *Flow ID*, and the information asset types relevant to each data flow could be found in the data flow mapping table that was created at the beginning of this stage. Note that we tagged all the data flows with ID and identified their relevant information assets, regardless of whether the flow is in the scope of our threat elicitation or not. We decided to document the full system data flows mainly for two reasons: First, for a better

[2] https://www.lucidchart.com/pages/data-flow-diagram/.

Table 3
List of Identified Information Assets.

Information Assets Category	Information Assets Sub-Category	Code	Description
Control Data		CD	Control commands
Operational Data		OP	Data representing the operational status of assets.
	Measurement Data	OP_MD	Data collected from multimeters.
	Alarm Data	OP_AL	Data representing alarm states and status codes
	Event Data	OP_EV	Data representing system events and respective status codes
Metering Data		MT	Data collected from energy meters and used for accounting and billing
Produced Data		PD	For example, processed measurement data, data produced by software applications, etc...
	Prediction Data	PD_PR	A Specific type of PD that is concerned with predicting measured or acquired values
	External Data	PD_ED	A specific type of PD that is acquired from microgrid external data sources (over the internet), (e.g., weather forecasts, electricity prices, etc)
Device Management Data		DM	Information about admin access to the devices, system patching, and device configuration
	Service Data	DM_SD	Admin access to devices for service and maintenance purposes
	System Logs	DM_SL	Device system logs
	Configuration Data	DM_CF	Device configurations

Table 4
Example of Entries in a Data Flow Mapping Table.

Flow ID	From	To	Connection Type	Information Asset Types
PV-08	PV Multimeter	**Server**	Digital Digital	OP DM
PV-09	PV Inverter Physical Process	**PV Multimeter**	Analog	OP_MD

Table 5
Default Data Flow Diagram Elements (Shostack, 2008).

Name	External Entity	Process	Data Flow	Data Store
Representation	Rectangular box	Circle	Directed arrow	Parallel lines
Definition	Things outside your control	Code	How information flows between other elements	Data at rest
Examples	People, other systems, web sites	.exe, assemblies, COM components	Function calls	Files, databases, registry keys

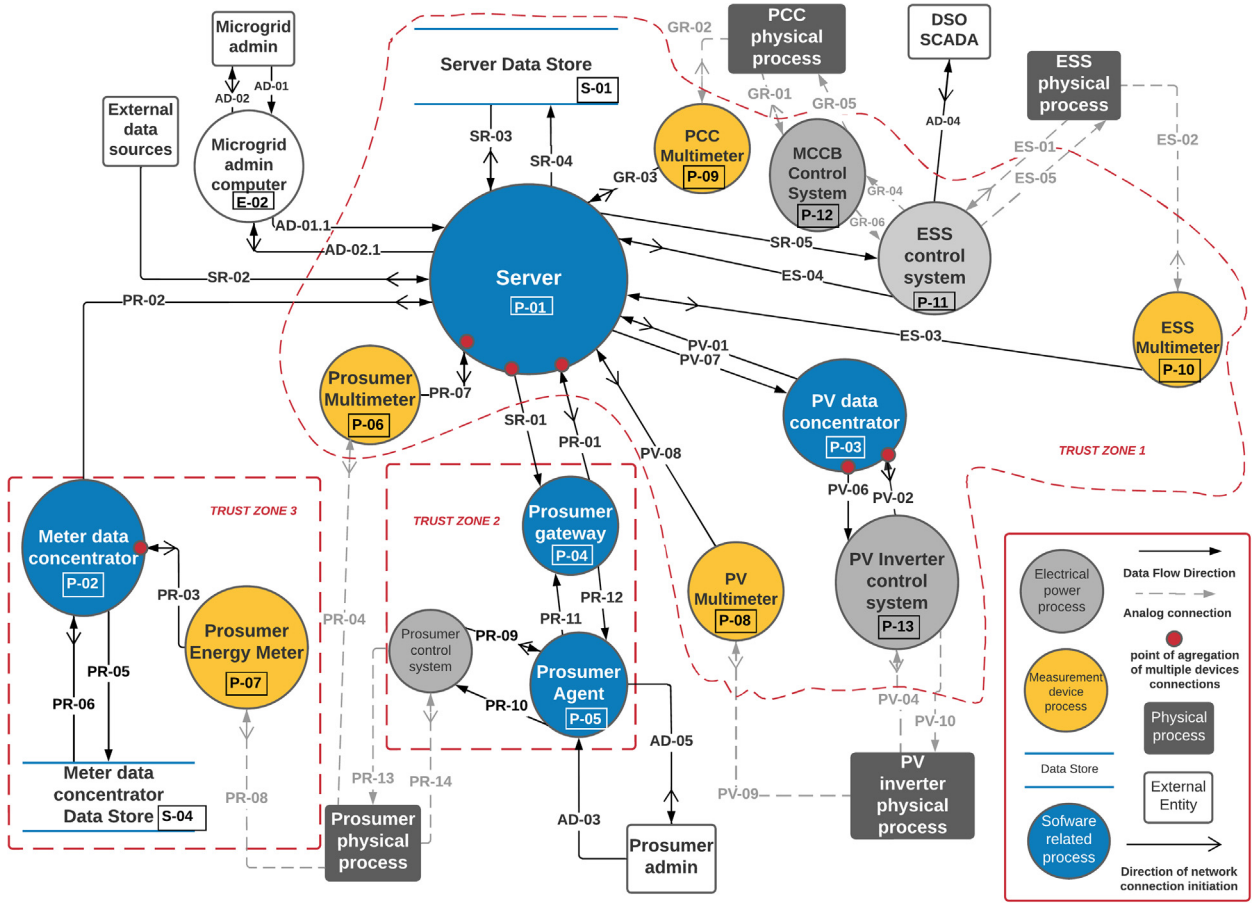

Fig. 5. Microgrid Data Flow Diagram.

understanding of the information assets and the data flows between all the system components for all the threat modeling contributors, especially contributors that do not have an electrical engineering background, like the cyber-security group. Second, as the system may change in time, there might be some flows that are excluded at this stage from the threat analysis, while they would be included in a future version of the threat-modeling, as each system modification should be questioned from the threat-modeling perspective. The limited number of system data flows made it feasible to map all data flows, while in more complex projects, it might be more challenging to keep track of analog data flows on the DFD, however, we still recommend documenting such information.

From the literature, we noticed that the DFD processes are usually presented in a single color. In the proposed DFD we mapped the four main types of processes discussed in Section 5.2 by a different color code for each process type, and following the same logic used in Fig. 3. For example, all measurement device processes (multi-meters and energy meters) are mapped by a yellow circle. On the other hand, the PLC processes, which are mainly related to electrical power, are mapped in gray, while the processes that are based on developed software are mapped using blue. Such differentiation made it easier for us during the threat elicitation phase to group processes with similar functionalities, as they tend to have similar information assets, which result in a high percentage of threat similarities.

Another challenge that we met during the creation of the DFD diagram was repeated components, as some processes are aggregating many connections coming from several devices that have the same type. For example, the meter data concentrator (P-02) aggregates information coming from several prosumer energy meters. We wanted to map such information on the DFD, as it would help us while thinking about the threats related to a device that has an aggregated link similar to the meter data concentrator. Mapping such aggregation with a single icon could also save space on the DFD, as it would substitute mapping each device several times. For this reason, we decided to assign a *red circle* that shows such aggregation on the DFD, as shown in Fig. 5.

For software-based systems, the external entities include people as well as systems, so an external system process might also be considered as an external entity. In our system, due to the assumptions discussed in Section 5.4, we have included the microgrid admin computer, as it should be possible for the microgrid admin to connect remotely and manage the system. As we are applying STRIDE per element, mapping the microgrid admin computer as an external entity, would force us to miss some threats, as the applicable threats on an entity are only the spoofing and repudiation threats. While threats related to a process apply all STRIDE aspects. As the microgrid admin computer is a computing device, we decided to treat it as a process. For this reason, we have mapped the microgrid admin computer as a new type of process that we expanded during the DFD creation, which is the *External Process* type, and which could not be trusted. We might also consider such an external process as a new element added to the DFD elements.

Regarding the identification of data stores, as per the definition provided in Table 5, these DFD elements correspond to system assets that host all kinds of data in rest, including files, databases, and registry keys. Cyber-physical system components represented as processes usually store information locally (e.g. registries and files), which should be considered as data stores. However, applying a strict definition of this element type may cause to have a distinct data store for each process, which may end up with a DFD with a very large number of elements. For this reason, we assumed that each process has information assets such as system logs and configuration data, which we identified as a sub-category of the device management data type, and we decided to not include dedicated data stores in the DFD for those information assets.

The data stores shown on the DFD were decided to be limited to the ones that host data such as measurement data, which are stored on the device for a long time (several hours or more) in any form (e.g., database, file). For this reason, the DFD contains only two data stores, even though each process can locally store data. The two identified data stores are locally available on their relevant physical device:Meter data concentrator data-store (S-04) is available on the meter data concentrator device, and the server data store (S-01) is a database that is hosted on the server. Mapping data stores in such a context helped us to identify where the sensitive information assets reside and to better assess the impact of the identified threats on the chosen information assets.

5.4. Security context

A system is considered as a part of an environment in security standards such as IEC 62443 (IEC, 2018) and Common Criteria (CCRA, 2018). Some security objectives may be fulfilled by the system while the remaining ones are left to the environment. Therefore, it is important to define the assumptions of the environment in addition to a clear definition of the system that will be located in this environment. The security context description should include the physical and logical location that a particular system will be deployed in, including the cyber and physical security countermeasures that should be provided by the environment (IEC, 2018).

From the onset as a precursor to any threat modeling process, of essence is a proper interpretation of the system under consideration in the form of a system model, which is realized by using DFD (discussed in Section 5.3). After developing a DFD, the usual practice is to discuss the trust boundaries, which leads to the main decision about including or excluding a DFD component in the threat elicitation phase. The assumptions, something accepted as true or factual, play a key role in boundary decisions. Although considerations about the assumptions are somehow inherent in such boundary discussions, we propose that threat modeling of CPSs can benefit from a dedicated stage for the discussion of assumptions among the team members.

It is important to clarify how physical and cyber environments interact with each other and which implications of such interactions should be reflected in the threat modeling study. Although we identified the cyber and physical assets and relevant analog/digital information flow in the DFDs, it is possible that a threat actor might physically access the devices or network and continue to launch malicious activities in cyberspace. Therefore, the assumptions about physical access should be considered thoroughly. Following discussion with the system owner, we agreed on the assumption that some system assets are physically secure, as they can be physically accessed only with authorized access. The identified physically secure system assets are the Microgrid server, ESS multimeter, PV data concentrator, PCC multimeter, MCCB Control System, ESS Control System, and PV Inverter Control System. The assumption included that these assets cannot be physically captured by a threat actor, either temporarily (in the short term) or continuously. We also assumed that the system owner physically protects the meter data concentrator, prosumer multimeter, and prosumer energy meter. As the prosumer gateway, prosumer agent, and prosumer control system are owned by the prosumer, we assumed that the prosumer physically protects these assets.

It was also important to understand the authorized physical and logical access to the different system components. We identified four operator groups for our microgrid system: *Microgrid admin, DSO admin, Prosumer admin*, and *Maintenance admins* (staff of the third parties such as device vendors). We assumed that microgrid and DSO admins are trusted entities, and do not have malicious

intentions, whereas prosumer admins and maintenance admins are not fully trusted, meaning that they can pursue some illegal goals. It is worth noting however that, the devices of all parties may be compromised or their credentials can be stolen. The last assumption resulted in mapping the microgrid admin computer as an external process, as we do not trust the computer, despite trusting the microgrid admin as a person.

The assumed capability range of the potential threat actors is a significant criterion for the determination of security requirements in IEC 62443. As detailed in Section 5.9, it is important to clarify which threat actors could be interested in attacking the modeled system and reflect this understanding while identifying the security requirements. In this context, we agreed on the assumption that nation-state actors may not get a significant political, economic or military advantage by compromising a microgrid system, thus, we eliminated this potential threat agent from our analysis. This assumption implies that the cyber-attacks requiring huge resources and advanced capabilities are out of our scope. Non-state actors such as terrorist groups who may utilize cyber means are not considered viable threat agents either. They intend to create a sudden shock in the whole society by inducing physical harm that can be visually demonstrated (Rekik et al., 2018). It is very highly unlikely for such a group to plan an attack on a small-scale industrial microgrid system. However, cyber-criminals can target such a system as energy could be a lucrative market for gaining financial outcomes. For instance, they may consider launching ransomware attacks to threaten the owners of the microgrid systems by inducing operational costs. Despite their lower-level technical capabilities, hacktivists may target microgrid systems for protesting energy-related policies or practices of microgrid owners. On the other side, we assume that prosumers may have malicious intentions to manipulate their energy production and consumption data to get an economic advantage. They can also launch attacks on other prosumers with the same purpose (e.g., having an advantage in an energy bidding).

Another assumption is that supply chain attacks that introduce hardware-based modifications to the microgrid components at the manufacturing stage (e.g., hardware-based trojans (Lingasubramanian et al., 2018)) are out of scope due to complexity. The attacks arising solely from the abuse of physical attack vectors (e.g., equipment theft, physically damaging devices, or a system) were excluded from the scope of our threat modeling. We have also excluded the physical attacks causing violations resulting in impairment of the sensing capabilities of the devices. While we agreed that our threat modeling analysis would include physical devices compromise, for devices that are located in insecure places, by accessing them via any logical and hardware interfaces and launching cyber-attacks (e.g., corruption of boot loader, deletion of the file system, modifying the critical data).

At last, some general clarifications about the system characteristics were made. The system owner clarified that the system has a purely wired communication network. As such, we excluded jamming and packet drop attacks (including attacks on time synchronization) at the threat elicitation stage. In addition, we found that the system does not generate, store or process sensitive personal information, thus we exclude privacy threats (we might have combined STRIDE with LINDDUN in this study, in case the system processes personal data).

The findings of this stage were documented at the beginning of the threat elicitation document, to make it clear for the reader in which security context the threat elicitation was done.

5.5. Trust boundaries

The trust boundaries were identified based on the assumptions discussed in Section 5.4. Some of these assumptions have a great impact on the chosen trust boundaries. For example, the assumption that several microgrid system components are physically secure and not to question network compromise attacks, and that the system does not include any type of wireless connection, resulted in a big trust boundary that we tagged as *Trust Zone 1* on the DFD shown in Fig. 5 (same components of the gray area shown in Fig. 3). Also, due to the microgrid site considerations, the PV devices (PV data concentrator, PV inverter control system, PV multimeter) were considered to be part of the microgrid *Trust Zone 1*, while this might be modified if the same system is implemented in another location in the future.

One of the sensitive decisions that were taken during the discussion of the assumptions, was the assumption that the prosumer energy meter is sealed and its network is trusted. Such an assumption resulted in including the prosumer energy meter in the same trust boundary as the meter data concentrator and its data store, which we tagged as *Trust Zone 3* in Fig. 5. This assumption caused the exclusion of many threats that are discussed in some studies regarding the security of prosumer energy meters (e.g., (Hoeve and Peters, 2019)). For instance, if we do not consider this assumption, prosumer with malicious purposes may launch false data injection attacks on PR-03 to lower his/her bills. On the other side, the data flow PR-02 between the meter data concentrator (P-02) and the Server (P-01) is not within this trusted zone, thus, the attackers can still compromise this flow.

The remaining trust boundary was decided based on the ownership, as we decided to include the prosumer control system, the prosumer gateway, and the prosumer agent in the same trust boundary (i.e., *Trust Zone 2*), as they are all owned by the prosumer.

As all physical processes and their analog connections are out of the scope of our analysis, the physical processes were mapped out of the system trust boundaries.

The identified trust boundaries might be modified during the different phases of the system lifecycle. For example, in case of any assumption modification or when the system is applied to a new site, this might result in totally different trust boundaries. For these reasons, it is important to document the factors on which the trust boundaries decisions were taken for each threat-modeling exercise run, so that it could be reviewed in future runs.

Table 6
Mapping STRIDE Threats to the Proposed Attack Taxonomy.

STRIDE Element	Precondition	Possible Attack Vector(s)
Spoofing		Spoofing Attacks
Tampering	Device Compromise	Integrity Attacks
Repudiation	or	Delete/Corrupt Data
Information Disclosure	Network Compromise	*Not applicable*
Denial of Service		Availability Attacks
Elevation of Privilege		Escalation of Privilege

5.6. Threat elicitation

During the threat elicitation stage, each threat definition generated for a specific DFD element has mainly three common components: (1) A precondition attack state that is derived from the attack taxonomy, (2) The possible attack vector(s) applicable for that threat, (3) The impacted information assets. To ease the integration between STRIDE and the proposed attack taxonomy in Fig. 4, we summarize the mapping of STRIDE to the first two components of the proposed threat definition (precondition attack, and possible attack vectors) based on the taxonomy.

As shown in Table 6, the precondition attack state is selected from the *Device* or *Network Compromise* categories of the attack taxonomy. Process or data store are assumed to be captured by

Table 7
A Sample Set of Threats.

DFD Element Type and Code	Process, P-02
DFD Element Name	Meter Data Concentrator
Information Assets	Operational Data, Metering Data, System Logs and Configuration Data
Spoofing Threat	A threat agent who ***compromises the network***, uses ***a spoofing attack*** to masquerade as a **Meter Data Concentrator**.
Tampering Threat	A threat agent who ***compromises the Meter Data Concentrator or misuses his/her legitimate*** access, uses ***tampering/forging attacks*** to manipulate/fake **operational data/metering data/system logs/configuration data in use**.
Repudiation Threat	A threat agent who ***compromises the Meter data concentrator (excluding physical compromise) or misuses his legitimate access, delete/corrupt*****the** system logs to cover tracks of his malicious activities.
Information Disclosure Threat	A threat agent who ***compromises the Meter Data Concentrator or misuses his/her legitimate access***, gets access to **system configuration and logs/operational data /metering data in use.**
Denial of Service	A threat agent who ***compromises the network*** uses ***a resource consumption attack*** to make the **operational data/metering data in use** unavailable.
Elevation of Privilege Threat	A threat agent who ***compromises the Meter Data Concentrator or misuses his/her legitimate access***, with limited access, ***escalates his/her access level to root access*** on the process P-02.

the *Device Compromise* category whereas an information flow is attackedby *Network Compromise*. If an entity that is assumed not to be fully trusted accesses a device (i.e., third-party maintenance staff), this precondition is grasped by the statement, *misuse his/her legitimate*. The attack vectors are mainly selected from the *Integrity* and *Availability Attacks* categories of the attack taxonomy, in addition of *Spoofing* and *Escalation of Privilege Attacks*. We assume that the Information Disclosure attacks can be easily realized after the compromise of a device or network, thus, we did not add a specific attack vector to the definitions of such threat category, instead, only the keyword, *access to*, is included in the threat definition. The information assets which are subject to the given attacks are listed in each threat. We differentiate the state of the information assets by the keywords, *in use* for a process, *in transit* for an information flow, and *at rest* for a data store. Table 7 provides a sample of the elicited threats of process P-02. For instance, in the tampering threat, the text, *compromises the Meter Data Concentrator or misuses his/her legitimate* specifies the precondition, *tampering/forging attacks* indicate the attack vector applied in the threat, *operational data/metering data/system logs/configuration data* determines the list of information assets impacted by this threat, and *in use* indicates the asset state. Exceptions to this threat definition notation are the threats regarding the categories, *Spoofing*, and *Elevation of Privilege*. Still, such threats have specific templates, we contemplate that these threats define the precondition attack, and attack vectors, however, the attack vector in these two types of threats is actually the initial attack stage for the follow-up attacks, as it does not have direct tangible harm on the assets. For this reason the threat definitions of Spoofing and Elevation of Privilege threats do not contain the impacted information asset component. For instance, the spoofing threat in Table 7 has the description, *a threat agent who compromises the network, uses a spoofing attack to masquerade as a Meter Data Concentrator*. Here, the spoofing threat fulfills the initial stage for further attacks (e.g., *Integrity Attacks*). However, still, it constitutes a distinct threat that should be considered while deriving the security requirements.

It is important to note that some descriptions (e.g., precondition descriptions such as "compromises the Meter Data Concentrator") seem to be so general as they use the name of the category (e.g., device compromise). This means all the corresponding attack subcategories given in the taxonomy can be a viable device compromise method (e.g., firmware compromise, malware, physical compromise). Thus, referring to the taxonomy during the stage of security requirement selection can inform the experts about more technical details which may be instrumental to identify the most convenient requirement alternative.

The threat elicitation stage resulted in the elicitation of 110 threats based on STRIDE, where 12 threats are related to spoofing, 23 threats related to tampering, 17 threats related to repudiation, 22 threats related to information disclosure, 26 threats related to denial of service, and 10 threats related to elevation of privilege. We elicited the threats related to 7 data flows, 13 processes, 2 data stores, 4 external entities, and an external process. The trusted data flows were not analyzed at this stage, while they might be analyzed in future runs of the threat-modeling exercise, in case of trust boundary modifications.

5.7. Threat consequences

Comprehending the consequences of the threats is a significant effort to assess and rank the threats. We identify the possible consequences of each elicited threat in the seventh stage, following the idea of Khan et al. (2017) in which each threat is mapped to threat consequences and then hazards. We used the term, loss, instead of hazard as the cyber security community is more familiar with it.

In software-based systems, the information is the main subject of the attack, thus, it is comparatively easy to predict the harm based on the violation of security properties, confidentiality, integrity, and availability. However, in cyber-physical systems, the consequences, especially the physical ones, cannot be deduced at the first glance and require more in-depth considerations from a team that is usually composed of security and system experts (e.g., power experts in the case of microgrids). The elicited threats retrieved at the end of the previous stage (i.e., threat elicitation) include definitions regarding the impact on information assets. The threat consequence stage complements the elicitation by mapping threats to the potential final harm including physical ones.

At this stage, we used two terms, *Threat Consequence* and *Loss* for defining the implications on the people or business processes. *Threat Consequence* briefly explains the immediate result of the corresponding threat, whereas *Loss* outlines the final harm of the threat. For instance, an integrity threat to control data of ESS system can induce that ESS Control system exceeds safe operation limits (i.e., a threat consequence definition), which may lead to loss of life or injury to people (i.e., a loss definition).

We identified potential threat consequences and loss definitions from MITRE's framework "ATT&CK for Industrial Control Systems" (i.e., we benefited from the definitions of impact) (Alexander et al., 2020) and the paper of Khan et al. (2017). The power system expert took the lead in the discussions of this stage, modified the initial list, and added new items. The final lists are given in Tables 8 and 9.

A sample mapping for the threat T_57, which is a tampering threat on the process ESS Control system, is presented in Table 10. The table includes the list of threat consequences of the threat T_57, and mentions the list of potential losses that might result from each threat consequence (TC). For example, TC-1 (overload or exceed operational limits) might result in loss of life or injury to people, loss or damage to property of micro-

Table 8
Threat Consequences Definitions and Priorities.

Code	Threat Consequence Definition	C	H	M	L
TC-1	Overload or exceeding operational limits	1	4	3	0
TC-2	Disconnection from the microgrid	0	2	3	0
TC-3	Interrupting the energy transfer	0	1	3	0
TC-4	Interrupting/Manipulating the control	0	0	1	0
TC-5	Inability to provide ancillary services	0	0	3	0
TC-6	Misleading safety operations	1	3	0	0
TC-7	Limited energy availability	0	1	3	0
TC-8	Misleading connection/disconnection operations	0	1	2	0
TC-9	Violation of power quality	0	3	1	0
TC-10	Insufficient power availability	0	1	3	0
TC-11	Loss of energy sold	0	0	3	0
TC-12	Insufficient visibility of inverters	0	1	2	0
TC-13	Leaking information for further attacks	0	0	0	1

Table 9
Loss Definitions and Prioritization.

Loss Code	Loss Definition	Priority
L-1	Loss of life or injury to people	Critical
L-2	Loss or damage to property of microgrid	High
L-3	Loss or damage to property of prosumer	High
L-4	Loss or damage to property of TSO/DSO	High
L-5	Loss of power generation	High
L-6	Loss of mission	Medium
L-7	Loss of view	Medium
L-8	Loss of customer satisfaction	Medium
L-9	Loss of information	Low
L-10	Loss of revenue	Medium

Table 10
A Sample Mapping of Threats to Consequences and Losses.

Threat	Threat Consequence	Loss
A threat agent who compromised the ESS control system or misuses his legitimate access, uses tampering/forging attacks to manipulate/fake control data/ operational data/device management data in use.	TC-1	L-1, L-2, L-3, L-4 L-5, L-6, L-8, L-10
	TC-2	L-3, L-5, L-6, L-8, L10
	TC-3	L-5, L-6, L-8, L-10
	TC-4	L-7
	TC-7	L-5, L-6, L-8, L-10
	TC-10	L-5, L-6, L-8, L-10
	TC-11	L-6, L-8, L-10

grid/prosumer/Transmission System Operator (TSO)/DSO, loss of power generation, loss of customer satisfaction, and loss of revenue.

5.8. Threats prioritization

Due to time and resource limitations, a decision might be taken to address only high-priority threats and accept the remaining ones in the security requirements selection phase.

Risk is widely defined by an equation of likelihood and impact. We contemplate that, in our case, we do not have enough information (i.e., system details such as OS, protocols, and product vendors) and solid criteria for calculating the likelihood of the threats at this early stage. Whereas conducting an impact assessment is still feasible and relevant, which could be done based on the identified consequences and expected losses as given in Section 5.7. For this reason, we decided to perform an impact assessment, instead of a risk assessment, to prioritize the threats in our case study. After initially assigning priority levels to losses, which was mainly done by the energy expert, such results are traced back to threat consequences and then threats. Thus, our impact assessment method consists of three stages: 1) Loss prioritization 2) Threat consequence prioritization 3) Threat prioritization and selection.

Table 11
Example of Impact Calculation of a Threat Consequence.

Threat Consequence Code	Loss	Priority
TC-1	L-1	C
	L-2	H
	L-3	H
	L-4	H
	L-5	H
	L-6	M
	L-8	M
	L-10	M
TC-1 impact		1 C 4 H 3 M 0 L

Table 12
Example of Threats Prioritization.

Threat Code	C (Critical)	H (High)	M (Medium)	L (Low)
T_97	2	15	24	1
T_82	2	12	23	0
T_67	2	11	21	0
T_51	1	12	19	0
T_46	1	10	21	0
T_60	0	10	23	0
T_3	0	6	21	0
T_10	0	0	3	1
T_2	0	0	0	1

5.8.1. Loss prioritization

The list of losses identified in the previous stage was discussed with the system owner, then for each loss that is represented by a code, a priority was assigned based on the project goals and priorities (see the column named *priority* in Table 9). Four levels of prioritization were used: *Critical* (C), which is a loss that is necessary to avoid; *High* (H) means it is highly desirable to avoid that loss; *Medium* (M) indicates a reasonable loss; whereas *Low* (L) might be a tolerable loss, depending on the system owner decision.

5.8.2. Threat consequences prioritization

We have identified the list of losses that are applicable for each consequence. For example, TC-1 (overload or exceeding operational limits) could result in losses L-1, L-2, L-3, L-4, L-5, L-6, L-8, and L-10. We represented the cumulative losses as the total count of each priority level. As an example, the cumulative loss of TC-1 is considered as 1 Critical + 4 High + 3 Medium, as shown in Table 11.

5.8.3. Threats prioritization and selection

The priority levels of threats are calculated using the prioritization of threat consequences as obtained in 5.8.2. For example, threat T_1 *A threat agent who compromised the network uses Integrity attacks to manipulate operational data/metering data in transit to the server*, which is a tampering threat related to the data flow PR-2. We identified TC-3, TC-4, TC-7, TC-8, TC-10, and TC-11 as threat consequences of this threat. Considering the priorities of these consequences mentioned in Table 8, the priority of T_1 is calculated based on the addition of corresponding priority level counts of each consequence which add up to 4H + 15 M. We applied the same for the 110 elicited threats, then we sorted them based on a sequential sort, where we first sort the threats that has critical priorities, following by other priorities, as shown in Table 12.

The proposed threat prioritization method is beneficial to keep track of critical threats (where the threat impact is necessary to avoid), despite an alternative method that replaces each priority with a numeric weight. However, the resulted numeric number would not keep track of the number of critical threat consequences, which is a major information when prioritizing CPSs threats. For this reason, we decided to not replace each priority with a weighted value, instead, we keep track of each priority level

Table 13
IEC-62443 Security Levels.

Security Level	Description
SL 1	Prevent the unauthorized disclosure of information via eavesdropping or casual exposure.
SL 2	Prevent the unauthorized disclosure of information to an entity actively searching for it using simple means with low resources, generic skills, and low motivation
SL 3	Prevent the unauthorized disclosure of information to an entity actively searching for it using sophisticated means with moderate resources, IACS specific skills, and moderate motivation
SL 4	Prevent the unauthorized disclosure of information to an entity actively searching for it using sophisticated means with extended resources, IACS specific skills, and high motivation

Table 14
Selected Security Requirements for Threat T_82.

Threat ID	Requirements
T_82	CR 1.2 - Software process and device identification and authentication CR 1.2 RE (1) - Unique identification and authentication CR 6.2 - Continous monitoring

and apply a sequential sort. Hence, the consequences with higher priorities have more influence on the ranking, which enabled us to avoid misleading results of weighted sum values.

5.9. Selection of security requirements

The selection of security requirements is an essential phase of threat modeling. At this stage, we used the security requirement list of the international standard, *Security for Industrial Automation and Control Systems* IEC 62443, as a reference (Part 4-2: Technical security requirements for IACS components).

The IEC 62443-4-2 document provides requirements for each system component and then fine-tunes them for four specific component types which are named as follows: *Software Application Requirements* (SAR), *Host Device Requirements* (HDR), *Embedded Device Requirements* (EDR) or *Network Device Requirements* (NDR), and in case the same requirement is applied for the four components types, then it is represented in the document as *Component Requirements* (CR). The requirements are derived from seven fundamental requirement categories: *Identification and authentication control, use control, system integrity, data confidentiality, restricted data flow, timely response to events*, and *resource availability*. IEC 62443 addresses four security levels (SL) based on the required means and resources of the attacker to achieve the corresponding attack, as clarified in Table 13. Based on the security context discussed in Section 5.4, we agreed to select requirements that are relevant to SL 1, SL 2, and SL 3, and exclude the SL 4 requirements, as they require an advanced level of sophistication and resources that are relevant to state-sponsored attacks, which is out of our scope.

Cyber security experts selected relevant security requirements for each of the selected threats and created an initial requirements list. Then, they gave an overview of the standard to the remaining members of the team and presented the proposed requirements. The whole team reviewed the selected requirements and discussed their feasibility. A sample of selected security requirements for the critical spoofing threat T_82 on the Server process, which is described as *A threat agent who uses spoofing attacks (related to PR-02) masquerades as the server*, is presented in Table 14. The selected requirement CR 1.2 and its enhancement RE (1) recommend that the server process identify itself and authenticate to any other component, which could prevent spoofing attacks. The requirement CR 6.2 recommends the use of continuous monitoring to detect IP spoofing through monitoring tools that can detect and report breaches promptly.

6. Discussion

This study proposes a systematic methodology to apply STRIDE on CPSs. The paper answers the research questions RQ(1) and RQ(1.a), by proposing a nine stage methodology, that was discussed in Section 3. We also provide details on how to apply each stage of the proposed methodology, and showed a real-world case study regarding the threat modeling of a microgrid system. The study answers the research question RQ(2) in Section 5.3, by proposing new elements to the traditional DFD (e.g., adding physical processes to the DFD, and the use of colors to differentiate the processes based on their functions). Such modifications provide a better visibility of the different system components and help in a better understanding of the system, and the DFD. The Section 5.6, provides an answer to RQ(2), by showing how to integrate STRIDE elements with the proposed attack taxonomy in a threat definition. We also answered RQ(2) by proposing threat prioritization through impact assessment, discussed in Section 5.8.2. Our threat prioritization method provides a better traceability of critical threats, as some CPS threats might result in life's losses and/or physical damages.

In our study, attack taxonomy constitutes a knowledge base that assists in the systematization of the threat elicitation process. We updated it in the light of team discussions and the knowledge acquired from other sources during the threat elicitation work. In typical organizational settings, a security team conducts several threat modeling studies for various projects. Maintenance of such a knowledge base can also aid in transferring experience among the projects.

We contemplate that the level of abstraction in this knowledge base is fit for purpose, as our study covers the conceptual and requirement analysis phases of a development endeavor whose core intention is to determine the main security functions before the system design. We identified the primary attack categories which are not further detailed according to the variations in the attack tactics and techniques. As we do not know the system details such as operating systems, network protocols, and specific network topology at this stage, we couldnot incorporate the information that can be obtained from more resources such as vulnerability databases. However, in the later stages of the development, it would be very beneficial to incorporate those information into the analysis.

Critical to the threat modeling process are assumptions that could be specific to the target system. They should be done carefully as they root out the threats deemed irrelevant, offering a focus on plausible threats. In cyber-physical systems, the threat landscape may cover various combinations of cyber and physical attacks (i.e., a categorization of such attacks for smart grid systems is given in (Pillitteri and Brewer, 2014)). Especially, the scope should be clarified regarding the physical attack vectors which can be used for the compromise of a system (e.g., physically access to the devices to achieve a precondition attack state) as well as inducing the final harm to the systems (e.g., physically destruct the target device or steal the device itself). In our case, we excluded the later attack category as we focus on the threats using cyber means. Note that we still address the cyber-attacks causing physical harm. However, we assume that the attackers can physically access some of the devices for conducting cyber-attacks.

The list of threats elicited based on STRIDE-per-element is limited to each system component, thus, the threats do not reflect the full attack scenario starting from the attacker's interaction with

the attack surface and ending with the actions regarding the actual harm. STRIDE does not model the lateral movements and their consequences. In our case, a threat agent who compromises the server through a phishing email, for example, would have the opportunity for further lateral movement to other system components (e.g., processes or data stores). To track such scenarios, it might be useful to complement STRIDE with methods such as attack trees, to show the whole attack path. In this way, different threats elicited from various DFD components can be analyzed holistically to achieve the purpose of an attack scenario. However, the present study does not include such an additional analysis stage which is left to the future work.

The proposed trust boundaries may not hamper such lateral moves within a specific boundary as communication between peers in the same trust boundary is trusted and allowed. On the other side, recently, the concept of zero trust (Buck et al., 2021) has drawn particular attention in the industry. A solution that applies this concept would treat each process/data store as a separate trust zone. While such an action would result in a more secure system, the implications, particularly at the threat elicitation stage, would be complex as the system would have a high number of trust zones. This would also result in a longer list of requirements that might be challenging to implement. We consider that implementation of zero trust is a critical design decision that may enormously favor security in the usual trade-off between functionality and security. Special attention should be given when the target is a CPS as such systems do not usually tolerate functional disruptions. In our case, we assumed that zero trust is not the goal of system design.

The study (Khan et al., 2017) proposes a methodology for the application of STRIDE which sets some initial starting points for our study. The paper provides an overview of the stages of a system mapping to DFD without going into details that support the application of the procedures in a real-world scenario. Also, the threat elicitation stage requires advanced knowledge to assess the threat consequences without having a list of potential threats. Hence, following the proposed methodology was challenging for our threat modeling team. For this reason, we proposed to perform the threat elicitation before the identification of threat consequences, to make it easier for the project team to brainstorm and come up with a list of threat consequences based on the elicited threats. Such challenges in applying the methodology proposed in (Khan et al., 2017) lead us to propose a new methodology that aims for improved completeness and ease of adoption. When comparing our case study to Khan et al. (2017), we introduced a more systematic threat elicitation procedure by defining an attack taxonomy and showed how to incorporate the information assets and attack vector(s) into a structured threat descriptions. We provided a detailed discussion about the assumptions, their reflections on the DFDs, and what their impact might be in the elicitation phase. Our study also proposes modifications to the traditional DFD to make it more relevant to CPS systems. The proposed methodology is complemented by stages in which threats are prioritized and the security requirements are selected. However, (Khan et al., 2017) mainly cover the stages 3 (System Mapping into DFD), 6 (Threat Elicitation), and 7 (Threat Consequences and Losses Identification) of the methodology proposed in Fig. 1.

Our threat modeling methodology is based on STRIDE which has been widely used in industry and academic literature, as clarified in Section 2. Some studies in the literature extend the safety framework, System-Theoretic Process Analysis (STPA) for cyber security (Friedberg et al., 2017; de Souza et al., 2020). Although the top-down approach of this framework (i.e., defining the control layers, deriving the hazardous control actions, and completing the relevant mappings) may provide a strong linkage between the cyber threats and their physical results, the modeling takes so much effort. The members of the threat modeling team, including the power and security experts, found it very difficult to identify and map the control loops in such a considerably complex system. This is the reason why we favored the bottom-up approach of STRIDE, in which the determination of threat consequences is left after identifying the systems and related threats.

Although we were able to predict the information/system assets in addition to rough network topology and necessary information flows of the prospective system, more technical details about the network protocols, OS types, or network devices were unknown. Therefore, our study seeks to identify the key security functions at this early stage rather than specific vulnerabilities of the system components. However, we acknowledge the need for a thorough re-assessment of threats in the later stages when more details of the system architecture become clear.

The threat modeling studies in the literature usually do not demonstrate how their results could be incorporated into a framework of a standard. In our case study, we used the requirement list of IEC-62443 at the stage of security requirement selection. As, in this standard, the attacker profile is the main variable for deciding the security level of the system, which has a major impact on the requirement selection, we provided assumptions that clearly explain which malicious actors are considered within the scope and which security level should be targeted.

Threat modeling is considered to have low-level maturity in terms of research and practice (Yskout et al., 2020). The studies usually lack the validation of the outcomes (Tuma et al., 2018) and the quality of the results can depend significantly on the experiences of the experts involved in the process (Yskout et al., 2020). Consequently, the validity of our study presents similar limitations, which we aimed to minimize as best as practicable, for example, by using published technical documents (Egozcue et al., 2012; EN/CENELEC/ETSI, 2019; Muhammet Oztemur, 2014; SMGW-PP, 2014; Suleiman et al., 2015) to cross-check our results after threat elicitation. A benefit of the proposed approach to incorporate an attack taxonomy into the threat elicitation has significant potential to reduce the reliance on expert knowledge to some extent. Nonetheless, systematic and provable validation that measures the impact of including the taxonomy on the quality of outcomes remains as a challengefor this area of research. This aspect constitutes one line of our future work.

It is important to note that, apart from the threat elicitation, expert knowledge is also required for identifying the target system, assets and relevant information flows. As the modeling is performed at the early stage of the development, there could be some missing information or misinterpreted issues that may have an impact on the outcomes. Although the system owner has actively participated in the discussions, still, the issues related to the perception of system details constitute an important factor in the overall validity.

We comtemplate that the proposed threat modeling method and workflow could be applied to other CPSs beyond microgrids as we do not utilize any construct that is specific to microgrids in any step. However, scalability would be a problem for systems having large number of elements, as we apply the STRIDE-per-element approach which necessitates analyzing each element. In response to this complexity, the decomposition of the whole system into subsystems can be determined at an abstraction level that may lead to less effort but still enable the analysts to identify significant threats at the early stage.

Nevertheless, practitioners can greatly benefit from systematically following the proposed threat modeling methodology and accumulating knowledge in the form of attack taxonomies. Despite the challenge in realizing provably valid threat modeling methods, our position is that creating a threat modeling process, rigorously following it in several projects, paying attention to systematically

improving the knowledge-base, and creating well-established communication channels among the stakeholders are the key factors in long-run success for secure systems.

7. Conclusion

This paper provides a threat modeling methodology that is supported by systematic asset identification, system modeling, and threat elicitation procedures for CPSs. A system and software development project that aims to support the effective management of a microgrid site constitutes our case study to show the applicability of our methodology. Although we utilized the system modeling method of STRIDE in the form of DFD and its main threat categories as a departing point for the threat elicitation, we provided a comprehensive threat modeling guideline that consists of nine stages for a cyber-physical system.

We explained in detail how we identified the system assets, discriminated the physical components from the cyber-related ones, determined the information assets, and revealed information flows. We elaborated on the interaction between cyber and physical spaces, deduce security assumptions and show how such assumptions can be handled in the DFDs. We give a detailed discussion about how we created the trust boundaries in the light of assumptions and system modeling. We created an attack taxonomy for a microgrid system and incorporated it into a systematic threat elicitation stage, which has been usually done ad-hoc in practice. This study provides a detailed example that could support practitioners and researchers in applying threat modeling to a CPS.

Data Availability

Threat Modeling of Cyber-Physical Systems - A Case Study of a Microgrid System

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Shaymaa Mamdouh Khalil: Conceptualization, Methodology, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Hayretdin Bahsi:** Supervision, Conceptualization, Methodology, Investigation, Writing – original draft, Writing – review & editing. **Henry Ochieng' Dola:** Conceptualization, Investigation, Writing – original draft. **Tarmo Korõtko:** Funding acquisition, Project administration, Investigation, Writing – original draft. **Kieran McLaughlin:** Writing – review & editing. **Vahur Kotkas:** Investigation, Software.

Acknowledgments

This work has been supported by the Estonian Ministry of Education and Research and the European Regional Development Fund (grant 2014–2020.4.01.20–0289).

References

Ahn, B., Kim, T., Smith, S.C., Youn, Y.-W., Ryu, M.-H., 2021. Security Threat Modeling for Power Transformers in Cyber-Physical Environments. 2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT) 1–5. doi:10.1109/ISGT49243.2021.9372271.

Alexander, O., Belisle, M., Steele, J., 2020. Mitre att&ck® for industrial control systems: Design and philosophy.

Bahsi, H., Dola, H.O., Khalil, S.M., Korotko, T., 2022. A Cyber Attack Taxonomy for Microgrid Systems. In: IEEE 17th Annual System of Systems Engineering Conference.

Buck, C., Olenberger, C., Schweizer, A., Völter, F., Eymann, T., 2021. Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust. Computers and Security 110, 102436. doi:10.1016/j.cose.2021.102436.

Canaan, B., Colicchio, B., Ould Abdeslam, D., 2020. Microgrid cyber-security: Review and challenges toward resilience. Applied Sciences 10 (16).

CCRA, 2018. Common criteria for information technology security evaluation part 1: Introduction and general model.

Egozcue, E., Rodríguez, D.H., Ortiz, J.A., Villar, V.F., Tarrafeta, L., 2012. Annex II. Security aspects of the smart grid. Enisa April, 71.

EN/CENELEC/ETSI, 2019. Protection Profile for Smart Meter - Minimum Security requirements. Technical Report.

Fernandez, E.B., 2016. Threat modeling in cyber-physical systems. In: 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing. IEEE, pp. 448–453.

Friedberg, I., McLaughlin, K., Smith, P., Laverty, D., Sezer, S., 2017. Stpa-safesec: Safety and security analysis for cyber-physical systems. Journal of information security and applications 34, 183–196.

Haider, M.H., Saleem, S.B., Rafaqat, J., Sabahat, N., 2019. Threat Modeling of Wireless Attacks on Advanced Metering Infrastructure. MACS 2019 - 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics, Proceedings doi:10.1109/MACS48846.2019.9024779.

Hoeve, M., Peters, C., 2019. Security requirements for procuring smart meters and data concentrators. ENCS 11, 32.

IEC, 2018. Security for industrial automation and control systems—part 4-1: Secure product development lifecycle requirements.

ISAGCA, 2020. Security Lifecycles in the ISA/IEC 62443 Series. Security of Industrial Automation and Control Systems(October), 1–18.

Jamil, A.-M., Othmane, L.b., Valani, A., 2021. Threat modeling of cyber-physical systems in practice. arXiv preprint arXiv:2103.04226.

Khan, R., Mclaughlin, K., Laverty, D., Sezer, S., 2017. STRIDE-based Threat Modeling for Cyber-Physical Systems, 0–5.

Korõtko, T., Rosin, A., Ahmadiahangar, R., 2019. Development of prosumer logical structure and object modeling. In: 2019 IEEE 13th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG), pp. 1–6. doi:10.1109/CPE.2019.8862390.

Lingasubramanian, K., Kumar, R., Gunti, N.B., Morris, T., 2018. Study of hardware trojans based security vulnerabilities in cyber physical systems. 2018 IEEE International Conference on Consumer Electronics, ICCE 2018 doi:10.1109/ICCE.2018.8326180.

Muhammet Oztemur, N.G., 2014. Common Criteria Protection Profile for Smart Meter of Turkish Electricity Advanced Metering Infrastructure. Technical Report. Turkish Standards Institution.

Pillitteri, V. Y., Brewer, T. L., 2014. Guidelines for smart grid cybersecurity.

Ramis Ferrer, B., Afolaranmi, S.O., Lastra, J.L.M., 2017. Principles and risk assessment of managing distributed ontologies hosted by embedded devices for controlling industrial systems. Proceedings IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society 2017-Janua, 3498–3505. doi:10.1109/IECON.2017.8216592.

Rekik, M., Chtourou, Z., Gransart, C., 2018. A Cyber-Physical Threat Analysis for Microgrids.

Scandariato, R., Wuyts, K., Joosen, W., 2015. A descriptive study of microsoft's threat modeling technique. Requirements Engineering 20 (2), 163–180.

Shostack, A., 2008. Experiences threat modeling at microsoft. CEUR Workshop Proceedings 413, 1–11.

Shostack, A., 2014. Threat modeling: Designing for security. John Wiley & Sons.

SMGW-PP, 2014. Protection Profile for the Gateway of a Smart Metering System. Technical Report. German Federal Office for Information Security.

de Souza, N.P., César, C.d.A.C., Bezerra, J.d.M., Hirata, C.M., 2020. Extending STPA with STRIDE to identify cybersecurity loss scenarios. Journal of Information Security and Applications 55. doi:10.1016/j.jisa.2020.102620.

Suleiman, H., Alqassem, I., Diabat, A., Arnautovic, E., Svetinovic, D., 2015. Integrated smart grid systems security threat model. Information Systems 53, 147–160.

Tuma, K., Calikli, G., Scandariato, R., 2018. Threat analysis of software systems: A systematic literature review. Elsevier, pp. 275–294.

Tuma, K., Scandariato, R., 2018. Two architectural threat analysis techniques compared. In: European Conference on Software Architecture (ECSA) doi:10.1007/978-3-030-00761-4_23.

Wuyts, K., Joosen, W., 2015. LINDDUN tutorial C(July).

Wuyts, K., Scandariato, R., Joosen, W., 2014. Empirical evaluation of a privacy-focused threat modeling methodology. Journal of Systems and Software 96, 122–138. doi:10.1016/j.jss.2014.05.075.

Wuyts, K., Van Landuyt, D., Sion, L., Joosen, W., 2010. LINDDUN: a privacy threat analysis framework.

Xiong, W., Lagerström, R., 2019. Threat modeling – a systematic literature review. Computers & Security 84, 53–69.

Yskout, K., Heyman, T., Van Landuyt, D., Sion, L., Wuyts, K., Joosen, W., 2020. Threat modeling: from infancy to maturity. In: 2020 IEEE/ACM 42nd International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER). IEEE, pp. 9–12.

Shaymaa Mamdouh Khalil is a PhD student and an early-stage researcher at the Center for Digital Forensics and Cyber Security at Tallinn University of Technology, Estonia. She had a BSc in electronics and communication engineering from Cairo University in 2006 and MBA International Paris from the University Paris Dauphine and the University Paris 1 Pantheon Sorbonne in 2013. In 2020 Shaymaa received her MSc cum laude in cyber security with a digital forensics specialization from

Tallinn University of Technology and the University of Tartu. Her research interests include cyber-physical systems security and digital forensics.

Hayretdin Bahsi is a research professor at the Center for Digital Forensics and Cyber Security at Tallinn University of Technology, Estonia. He has two decades of professional and academic experience in cybersecurity. He received his PhD from Sabancı University (Turkey) in 2010. He was involved in many R&D and consultancy projects about cybersecurity as a researcher, consultant, trainer, project manager, and program coordinator at the National Cyber Security Research Institute of Turkey between 2000 and 2014. His research interests include the application of machine learning to cyber security problems, digital forensics, and cyber-physical system security.

Henry Dola serves at TalTech Centre for Digital Forensics and Cyber Security. He was awarded a BSc in Information Technology from KCA University, Kenya in 2014 and the MSc degree in Cybersecurity from Tallinn University of Technology in 2021. His interests include network security and resilience of interconnected systems.

Tarmo Korõtko received the B.Sc. and M.Sc. degrees in mechatronics and the PhD. degree in energy and geotechnology from the Tallinn University of Technology (TUT), Tallinn, Estonia, in 2007, 2010, and 2019, respectively. In 2018, he became a member of the Microgrids and Metrology Research Group of the Department of Electrical Power Engineering and Mechatronics at TUT, where he is currently employed as a Research Scientist. He has published more than 15 articles on the topics of microgrid control, electric power system digitalization, local energy markets and communities, energy storage systems, and machine learning applications in electric power systems. His research interests include microgrids, local energy markets and communities, prosumers, power system digitalization, and artificial intelligence in electric power systems. He is a member of IEEE.

Dr. Kieran McLaughlin is a Reader at Queen's University Belfast, at the Centre for Secure Information Technologies (CSIT). He leads research in cyber security of smart grids, ICS, SCADA, and related cyberphysical systems. His research interests include threat analysis and intrusion detection for ICS/OT networking protocols, devices such as PLCs, digital twins in cyber security contexts, and the application of machine learning for automated intrusion response. He has been an investigator in several EPSRC and H2020 projects, and is a member of the UK's Research Institute in Trustworthy Inter-connected Cyberphysical Systems (RITICS), with three projects investigating secure smart grids and critical infrastructure.

Vahur Kotkas has been working on software architecture, model-based software development, logicbased program synthesis, and artificial intelligence topics for almost 30 years. During this time, he has created several software solutions both for academic and industry projects. One of his passions is network management and cyber security with the aim of simulation tools development. Vahur is the head of Applied Artificial Intelligence Working Group at Tallinn University of Technology, which focuses on the creation of logic-based artificial intelligence solutions that can be used in practice.