DOCTORAL THESIS

Threat Modeling for Industrial Control Systems: From System Design to Security Assessment

Shaymaa Mamdouh Mohammed Radwan Khalil

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY
TALLINN 2026

TALLINN UNIVERSITY OF TECHNOLOGY
DOCTORAL THESIS
83/2026

Threat Modeling for Industrial Control Systems: From System Design to Security Assessment

SHAYMAA MAMDOUH MOHAMMED RADWAN KHALIL

TAL TECH PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Department of Software Science

The dissertation was accepted for the defence of the degree of Doctor of Philosophy in Information and Communication Technology on 04/11/2026.

Supervisor: Research Professor Dr. Hayretdin Bahşi,
Department of Software Science,
School of Information Technologies,
Tallinn University of Technology,
Tallinn, Estonia

Opponents: Professor Dr. Sokratis Katsikas,
Department of Information Security and Communication Technology,
Norwegian University of Science and Technology (NTNU),
Gjøvik, Norway

Professor Dr. Jianying Zhou,
iTrust Centre for Research in Cyber Security,
Singapore University of Technology and Design (SUTD),
Singapore, Singapore

Defence of the thesis: 15/12/2026, Tallinn

Declaration:
Hereby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology has not been submitted for doctoral or equivalent academic degree.

Shaymaa Mamdouh Mohammed Radwan Khalil

---
signature

Copyright: Shaymaa Mamdouh Mohammed Radwan Khalil, 2026
ISSN 2585-6898 (paperback)
ISBN 978-9916-80-605-0 (paperback)
ISSN 2585-6901 (PDF)
ISBN 978-9916-80-606-7 (PDF)
https://doi.org/10.23658/taltech.83/2026
Printed by Koopia Niini & Rauam

Khalil, S. M. M. R. (2026). *Threat Modeling for Industrial Control Systems: From System Design to Security Assessment* [TalTech Press]. https://doi.org/10.23658/taltech.83/2026

TALLINNA TEHNIKAÜLIKOOL
DOKTORITÖÖ
83/2026

Ohtude modelleerimine tehnojuhtimissüsteemidele: süsteemi kavandamisest turvahindamiseni

SHAYMAA MAMDOUH MOHAMMED RADWAN KHALIL

TAL TECH
KIRJASTUS

Contents

List of Publications 8
Other Publications 8

Author's Contributions to the Publications 9

Abbreviations 10

1 Introduction 11
1.1 Research Questions 12
1.2 Thesis Structure 14

2 Background and Foundations of ICS Security 15
2.1 Security Challenges in ICS 15
2.1.1 An Expanded Attack Surface 15
2.1.2 Physical Consequences of Cyber Incidents 15
2.1.3 The Conflict Between Availability and Security 15
2.1.4 ICS Lifecycle and the Importance of Security by Design 16
2.2 Threat Modeling in the ICS Security Lifecycle 16
2.3 Relevant Standards and Frameworks 16
2.3.1 ISA/IEC 62443 16
2.3.2 MITRE ATT&CK for ICS 17
2.4 Summary 17

3 Methodology 18
3.1 Research Methods 18
3.2 DSR Cycle 1: A Structured Threat Modeling Methodology for CPS 20
3.3 DSR Cycle 2: Bridging Threat Modeling and Penetration Testing 21
3.4 Case Study on Threat Modeling and Cyber Resiliency Assessment Integration 21
3.5 Communication 21
3.6 Summary 22

4 Threat Modeling in Industrial Control Systems: State of the Art 23
4.1 Review Process and Selected Papers 23
4.2 A Taxonomy for ICS Threat Modeling 24
4.3 Threat Modeling Methods and System Mapping Techniques 25
4.4 Coverage of Threat Modeling Stages 26
4.5 Research Gaps 28
4.6 Summary 30

5 Threat Modeling Methodology for Cyber-Physical Systems 31
5.1 Methodology Overview and Design Rationale 31
5.2 AC Microgrid Case Study 32
5.2.1 Stage 1: Initial Attack Taxonomy Creation 32
5.2.2 Stage 2: Information and System Assets Identification 33
5.2.3 Stage 3: System Mapping into DFD 34
5.2.4 Stages 4 and 5: Security Context Definition and Trust Boundaries Determination 35
5.2.5 Stage 6: Threat Elicitation and Attack Taxonomy Update 36
5.2.6 Stage 7: Threat Consequences and Losses Identification 36

5.2.7 Stage 8: Threat Prioritization 37
5.2.8 Stage 9: Security Requirements Selection 37
5.3 Key Findings 38
5.4 Summary 39

6 From Design to Security Testing: Threat Modeling-Driven Penetration Testing of Newly Deployed ICS 40
6.1 Motivation and Design Rationale 40
6.2 The BRIDGE Methodology 41
6.3 Microgrid Case Study 43
6.3.1 DFD Update and Scoping 43
6.3.2 Test Planning: STRIDE to MITRE ATT&CK for ICS Mapping 44
6.3.3 Vulnerability Ranking: CVSS v3.1 vs. CVSS v4.0 44
6.3.4 Validation Against Design-Phase Outputs 45
6.4 Key Findings 45
6.5 Summary 47

7 Integrating Threat Modeling with Cyber Resiliency Assessment for ICS 48
7.1 Motivation 48
7.2 Conceptual Background 48
7.3 Framework Selection 49
7.4 The CRUC Methodology 49
7.5 Case Study 50
7.5.1 Context and Testbed 50
7.5.2 Operationalization 51
7.5.3 The Appropriate Level of Effort for Threat Modeling 53
7.6 Key Findings on Boundary Conditions 53
7.7 Summary 54

8 Discussion 55
8.1 Synthesis of Findings 55
8.2 Theoretical Contributions 56
8.3 Practical Implications 57
8.4 Limitations 59

9 Conclusion and Future Work 61
9.1 Conclusions 61
9.2 Future Work 61

List of Figures 64

List of Tables 65

References 66

Acknowledgements 71

Abstract 73

Kokkuvõte 74

Appendix 1 75
Appendix 2 97
Appendix 3 107
Appendix 4 125
Appendix 5 151
Appendix 6 173
Curriculum Vitae 181
Elulookirjeldus 185

List of Publications

The present PhD thesis is based on the following publications that are referred to in the text by Roman numerals.

I S. M. Khalil, H. Bahsi, and T. Korõtko. Threat modeling of industrial control systems: A systematic literature review. *Computers & Security*, 136:103543, 2024

II H. Bahsi, H. O. Dola, S. M. Khalil, and T. Korõtko. A cyber attack taxonomy for microgrid systems. In *2022 17th Annual System of Systems Engineering Conference (SOSE)*, pages 324–331, 2022

III S. M. Khalil, H. Bahsi, H. O. Dola, T. Korõtko, K. McLaughlin, and V. Kotkas. Threat modeling of cyber-physical systems - a case study of a microgrid system. *Computers & Security*, 124:102950, 2023

IV S. M. Khalil, H. Bahsi, and T. Korõtko. Penetration testing of newly-deployed industrial control systems. In J. Staggs and S. Shenoi, editors, *Critical Infrastructure Protection XIX: 19th IFIP WG 11.10 International Conference, ICCIP 2025, Revised Selected Papers*, volume 763 of *IFIP Advances in Information and Communication Technology*, pages 25–48, Cham, 2026. Springer

V S. M. Khalil, H. Bahsi, and T. Korõtko. Bridging industrial control systems design and testing through threat modeling-driven penetration testing – a microgrid case study. *Computers & Security*, 167:104906, 2026

VI T. Korõtko, S. M. Khalil, A. Schuller, V. Kotkas, J. Lambot, and K. Pappel. Implementation of MITRE CRUC for operational resilience assessment in cyber-physical systems. In *2026 IEEE International Conference on Industrial Technology (ICIT2026)*, 2026

Other Publications

The publications listed below were produced during the doctoral studies; however, they are not directly related to the main contributions of this thesis.

I S. Ali, A. Chub, H. Bahsi, and S. M. Khalil. Extending cyber threat modeling with physical dependency analysis in DC microgrids. In *2025 IEEE 66th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)*, pages 1–10, 2025

II M. Erbas, S. M. Khalil, and L. Tsiopoulos. Systematic literature review of threat modeling and risk assessment in ship cybersecurity. *Ocean Engineering*, 306:118059, 2024

Author's Contributions to the Publications

I In I, I was the main author. I contributed to the conceptualization and design of the study, conducted the literature review, analyzed the selected studies, prepared the figures, and wrote the manuscript.

II In II, I contributed to the development of the microgrid attack taxonomy through literature review and conceptual analysis, and reviewed the manuscript.

III In III, I was the main author. I contributed to the conceptualization and development of the threat modeling methodology, led the threat modeling sessions, documented and consolidated the results, led the requirements selection and prioritization process, including the prioritization calculations, prepared the figures, and led the writing, structuring, and comprehensive revision of the manuscript to ensure a consistent and coherent presentation.

IV In IV, I was the main author. I led the conceptualization and initial development of an approach linking threat modeling to penetration testing, contributed to the definition of the penetration test scope and planning based on the threat modeling results from the design phase, analyzed the penetration testing report, and wrote the manuscript.

V In V, I was the main author. This work extends the previous conference publication by fully developing and formalizing the proposed approach as the BRIDGE methodology, including its structure, detailed process, and empirical evaluation. I was responsible for conceptualizing and developing the methodology, analyzing the penetration testing report, preparing the figures, and writing the manuscript.

VI In VI, I contributed to the literature review and analysis of cyber resiliency assessment methods, supported the selection and understanding of the chosen cyber resiliency assessment approach, and contributed to structuring the application of the approach, including methodology steps, metrics selection, and mission definition. I also contributed to the sections related to threat modeling and background, and participated in team discussions and manuscript review.

Abbreviations

AC	Alternating Current
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
BRIDGE	Bridging Requirements and Implementation through Design-Guided Evaluation
CPS	Cyber-Physical System
CREF	Cyber Resiliency Engineering Framework
CRUC	Cyber Resiliency Use Case
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
DC	Direct Current
DFD	Data Flow Diagram
DSO	Distribution System Operator
DSR	Design Science Research
ESS	Energy Storage System
FMECA	Failure Modes, Effects, and Criticality Analysis
GMLC	Grid Modernization Laboratory Consortium
IACS	Industrial Automation and Control System
ICS	Industrial Control System
ICT	Information and Communication Technology
IEC	International Electrotechnical Commission
IoT	Internet of Things
ISA	International Society of Automation
IT	Information Technology
LINDDUN	Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, Non-compliance
MCCB	Moulded Case Circuit Breaker
MCDA	Multi-Criteria Decision Analysis
NIST	National Institute of Standards and Technology
OCTAVE	Operationally Critical Threat, Asset, and Vulnerability Evaluation
OT	Operational Technology
PASTA	Process for Attack Simulation and Threat Analysis
PLC	Programmable Logic Controller
PV	Photovoltaic
RQ	Research Question
SCADA	Supervisory Control and Data Acquisition
SDLC	Software Development Life Cycle
SL	Security Level
SLR	Systematic Literature Review
SQUARE	Security Quality Requirements Engineering
SSM-CR	Situated Scoring Methodology for Cyber Resiliency
STPA	Systems-Theoretic Process Analysis
STRIDE	Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege
TTP	Tactics, Techniques, and Procedures

1 Introduction

Industrial Control Systems (ICS) and Cyber-Physical Systems (CPS) play a key role in modern society, supporting essential services across critical infrastructure sectors. ICS are a specific class of CPS in which computational components directly monitor and control physical industrial processes such as water treatment, power distribution, and industrial manufacturing. While CPS represent a broader category encompassing a wide range of systems, including autonomous vehicles, smart buildings, and medical devices, ICS constitute the subset typically associated with critical infrastructure [49].

The integration of Information Technology (IT) and Operational Technology (OT) in ICS has enabled advances in automation, remote monitoring, and operational efficiency. However, this convergence has also expanded the attack surface and introduced new cybersecurity threats. The Ukrainian power grid attacks of 2015 and 2016 demonstrated how targeted cyber attacks on ICS can cause large-scale service disruption [32, 13]. The potential physical impact in ICS extends threat analysis beyond traditional considerations of data confidentiality, integrity, and availability to also encompass the consequences of cyber attacks on physical operations. Ensuring cybersecurity in ICS requires structured approaches that differ substantially from conventional IT security practices and capture the interactions between cyber and physical domains.

Threat modeling is a structured approach for identifying and mitigating a system's security threats across its lifecycle, with particular emphasis on the design phase, where addressing identified threats is considerably more effective and less costly than after deployment [21, 48]. When applied to ICS, however, its practical implementation presents significant challenges. A Systematic Literature Review (SLR) [25] conducted as part of this research revealed that, despite growing interest in threat modeling for ICS, existing approaches suffer from several limitations. First, although commonly used threat enumeration techniques such as STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege) [21, 48] are well established, there is no well-documented procedure for applying them in ICS environments and no common format for describing the elicited threats, leading to variations in how these techniques are applied and documented across ICS threat modeling studies. Second, the majority of the reviewed threat modeling studies focus primarily on threat identification without providing clear guidance for other stages of the process, such as threat prioritization and security requirements selection. Third, real-world case studies demonstrating the practical application of threat modeling remain scarce. Beyond these gaps, the outcomes of threat modeling are rarely linked to subsequent security assessment activities, such as penetration testing, leaving a gap between the design phase and security verification and validation testing. Unlike IT penetration testing, ICS penetration testing may trigger physical consequences during test execution, including safety incidents, which makes structured planning of each test a necessary precondition rather than an optional step [49, 26].

To address these challenges, this thesis proposes a CPS threat modeling methodology, grounded in a real-world microgrid case study, that supports the systematic identification and prioritization of security threats during the system design phase and derives security requirements for the most critical threats. The outcomes of the threat modeling process are then used to guide the planning of penetration testing for the same system, bridging the gap between system design and security testing. The thesis also investigates how threat modeling can be integrated with the MITRE Cyber Resiliency Use Case (CRUC) methodology [9] in ICS, revealing that CRUC does not provide sufficient guidance for conducting its threat modeling step in a systematic way.

1.1 Research Questions

This thesis is guided by four Research Questions (RQ) that collectively address the use of threat modeling across the ICS security lifecycle. Although ICS are the primary application domain of this thesis, given their critical infrastructure role, RQ2 is framed at the broader CPS level. The threat modeling methodology developed for RQ2 is designed for systems in which cyber and physical components interact, a characteristic that ICS share with other CPS such as autonomous vehicles, medical devices, and building automation systems. However, its application in this thesis is limited to an ICS case study. The questions were formulated in response to the limitations identified in the existing literature and reflect a progression from understanding the current state of the art, through the development and application of a structured threat modeling methodology for CPS and its extension to penetration testing, to the investigation of how threat modeling can be integrated with cyber resiliency assessment frameworks in practice. Each research question is presented below, together with a brief description of how it is addressed.

RQ1: *What is the state of the art in threat modeling for ICS, and what are the key limitations and research gaps in existing approaches?*

This question is addressed through a systematic literature review, which analyzes existing threat modeling approaches in ICS. The review identifies key characteristics of proposed methods, their coverage of threat modeling stages, and the main research gaps, particularly with respect to methodological clarity and validation.

RQ2: *How can threat modeling be systematically structured and applied to CPS, considering both cyber and physical aspects?*

This question focuses on the development of a threat modeling methodology tailored to ICS and CPS environments. It considers how commonly used techniques can be adapted to capture the dual nature of these systems: the cyber domain, where attacks originate and propagate, and the physical domain, where the consequences of those attacks manifest as operational disruptions, safety incidents, or physical damage. The goal is to provide a practical and systematic approach that can be applied by multidisciplinary teams in real-world settings, producing outputs that are meaningful to both cybersecurity specialists and domain engineers responsible for the physical processes.

RQ3: *How can the outcomes of design-phase threat modeling guide the penetration test planning and vulnerability assessment of newly deployed industrial control systems?*

This question investigates how the results of threat modeling can be used beyond the design phase to guide practical security assessment activities. In particular, it explores how identified threats can be translated into penetration test scoping and planning, enabling a more targeted and system-specific evaluation of security weaknesses in newly deployed systems.

RQ4: *How can threat modeling be integrated with cyber resiliency assessment frameworks in ICS, and what practical challenges arise when operationalizing this integration?*

This question examines how threat modeling can be integrated with cyber resiliency assessment frameworks in ICS. CRUC [9] was identified as a particularly relevant framework for this investigation as it explicitly includes a threat modeling stage. The

research explores the practical challenges that arise in operationalizing this integration, including the adequacy of framework guidance for conducting threat modeling in a systematic way.

RQ1, RQ2, and RQ3 are addressed following a Design Science Research (DSR) approach, organized as two consecutive cycles. RQ1 is addressed through a systematic literature review conducted as the first phase of DSR Cycle 1, and RQ2 and RQ3 are each addressed through their respective DSR cycle, as described in Sections 3.2 and 3.3 of Chapter 3. RQ4 is addressed through an exploratory case study. Table 1 maps each research question to the corresponding thesis chapter, publications, and contribution.

Table 1: Mapping of research questions to thesis chapters, publications, and contributions.

RQ	Chapter	Publications	Contribution
RQ1	4	I	1
RQ2	5	II, III	2
RQ3	6	IV, V	3
RQ4	7	VI	4

The thesis makes the following contributions to the field of ICS and CPS security:

1. A taxonomy of ICS threat modeling methods and a set of six threat modeling stages, used to analyze the existing literature in a systematic literature review [25]. The analysis of the stages covered by each study identifies seven research gaps, including the lack of structured and repeatable procedures, limited coverage of the stages beyond system mapping and threat enumeration, and the scarcity of real-world applications.

2. A nine-stage threat modeling methodology for CPS [24] that adapts STRIDE to cyber-physical environments. It includes procedures for asset identification and trust boundary determination, an extended DFD notation that represents the interaction between the cyber and physical domains, a threat elicitation procedure based on a domain-specific attack taxonomy [8], an impact-based threat prioritization procedure, and the selection of security requirements from IEC 62443. The methodology was demonstrated on a real-world Alternating Current (AC) microgrid during its design phase.

3. The BRIDGE methodology (Bridging Requirements and Implementation through Design-Guided Evaluation) [26], a nine-stage threat modeling-driven penetration testing methodology for newly deployed ICS, together with a mapping of STRIDE threat categories to MITRE ATT&CK for ICS techniques for structured test planning. Its application to the same microgrid provides a real-world ICS penetration testing case study, shows how test results can be traced back to the design-phase security requirements, and compares CVSS v3.1 and CVSS v4.0 for ranking ICS vulnerabilities.

4. An analysis of the boundary conditions for integrating threat modeling with cyber resiliency assessment, based on the operationalization of the MITRE CRUC methodology [9] on a manufacturing testbed [30]. The study shows that CRUC does not provide sufficient guidance for producing its threat model component and proposes a distinction between threat modeling-driven and threat modeling-based integration approaches.

1.2 Thesis Structure

This thesis is organized as follows. Chapter 2 provides the background and conceptual foundations of ICS security, introducing the key security challenges in ICS environments, the role of threat modeling in the ICS security lifecycle, and the two standards and frameworks used in this research: IEC 62443 and MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) for ICS. Chapter 3 presents the research methodology, describing the two DSR cycles and the exploratory case study adopted in this thesis and how they are applied across the different research activities. Chapter 4 presents a systematic analysis of existing threat modeling approaches in ICS, based on a systematic literature review, and identifies the key limitations and research gaps that motivate the contributions of this thesis. Chapter 5 develops and applies a structured threat modeling methodology to a real-world AC microgrid during its design phase, demonstrating systematic threat identification, analysis, and prioritization. Chapter 6 introduces the BRIDGE methodology and demonstrates how design-phase threat modeling outputs can be used to guide targeted penetration testing of newly deployed ICS. Chapter 7 investigates how threat modeling can be integrated with cyber resiliency assessment frameworks in ICS, examining the practical challenges of this integration through an application of the MITRE CRUC methodology. Chapter 8 synthesizes the findings across all research questions, discusses their theoretical and practical implications, and reflects on the limitations of the work. Finally, Chapter 9 concludes the thesis and outlines directions for future research.

2 Background and Foundations of ICS Security

This chapter introduces the foundational concepts on which this thesis is built. It covers ICS security challenges, the role of threat modeling in the ICS security lifecycle, and the two key standards and frameworks used in this research: IEC 62443 [1, 2] and MITRE ATT&CK for ICS [3].

2.1 Security Challenges in ICS

Securing ICS differs substantially from securing conventional IT environments, due to the operational nature of industrial systems, the physical processes they control, and the constraints that govern how they can be modified and tested. This section addresses four challenges directly relevant to this research: the expanded attack surface from IT and OT integration, the physical consequences of cyber incidents, the conflict between operational availability and IT security practices, and the long operational lifecycle of ICS that makes security by design a fundamental requirement rather than an optional consideration.

2.1.1 An Expanded Attack Surface

The growing connection between IT and OT has significantly expanded the attack surface of ICS. Systems that were once physically isolated are now connected to enterprise networks and, in some cases, accessible from the internet [49]. The attack surface of a modern ICS includes both IT components, such as servers and web applications, and OT components, such as field devices and embedded control units, each of which may present different types of vulnerabilities. The wide variety of device types, vendors, operating systems, and communication protocols within the same system makes a thorough security assessment considerably more complex than in a typical IT environment [25].

2.1.2 Physical Consequences of Cyber Incidents

In conventional IT systems, the impact of a cyber attack is typically bounded by the cyber domain, resulting in data loss, service disruption, or financial damage. In ICS, the same attack may propagate beyond the cyber domain and cause direct consequences in the physical domain, including equipment damage or harm to human safety [49]. This characteristic has a direct implication for how threats must be analyzed in ICS environments. It is not sufficient to assess what an attacker can do to a software component or a communication channel. The analysis must also trace what that action causes in the physical process connected to it. A threat that appears low in severity from a cyber perspective may become critical once its physical domain consequences are considered [49].

2.1.3 The Conflict Between Availability and Security

Industrial processes often run continuously, and any delay in control signals, whether caused by an attack or by an added security control, can push a physical process outside its safe operating range, resulting in serious consequences for system safety [49]. This fundamental characteristic creates a conflict between ICS operational requirements and standard IT security practices. Penetration testing illustrates this conflict clearly. In IT environments, a disruption caused by penetration testing is typically bounded by the cyber domain, resulting at most in temporary service degradation. In ICS, a test that causes a control command to be dropped, delayed, or modified may propagate beyond the cyber domain and trigger physical consequences that are difficult or impossible to reverse. For ICS that cannot tolerate any delay in control signals, even active network scanning may

cause unexpected device behavior or process disruption, requiring organizations to rely on passive or non-intrusive assessment methods instead [49]. Penetration testing of ICS, therefore, cannot follow the same unconstrained approach used in IT environments. Each test must be carefully scoped and planned to avoid unintended safety impact [25].

2.1.4 ICS Lifecycle and the Importance of Security by Design

The lifecycle of an ICS often exceeds ten years [49]. During this period, the threat landscape evolves continuously, new attack techniques emerge, and components that were secure at deployment will become vulnerable long before the system reaches the end of its operational life. As a result, systems deployed without a solid security baseline can accumulate significant security debt. Achieving an acceptable level of security at deployment is therefore essential. Unlike IT systems, ICS cannot easily be modified after deployment due to the cost of changes, the possibility of disrupting continuous operations, and certification requirements that restrict modifications to safety-critical devices [49]. Security requirements must therefore be identified and incorporated during the design phase, as these constraints make later changes significantly more difficult and costly to implement.

2.2 Threat Modeling in the ICS Security Lifecycle

Threat modeling is a structured process for analyzing potential security threats to a system. It was first developed in software security as a mechanism for identifying security problems during the design phase, before they are built into the system and become more costly to address [25, 21]. Addressing threats at this stage is considerably more effective than identifying them after deployment, when design decisions have already been implemented and are difficult to reverse.

STRIDE is a widely used threat enumeration technique in threat modeling, categorizing threats into six types: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege [21, 48]. STRIDE is typically applied to a system representation called a Data Flow Diagram (DFD), which maps the system's components, data flows, and trust boundaries in a way that makes potential threats visible and traceable.

Applying threat modeling in ICS environments introduces distinct challenges. The multidisciplinary nature of ICS, which brings together expertise from cybersecurity and various engineering disciplines, requires a process that is structured clearly enough to support participants from different technical backgrounds. Furthermore, as discussed in Section 2.1, threats in ICS must be evaluated not only for their impact in the cyber domain but also for their consequences in the physical domain. These characteristics mean that threat modeling approaches developed primarily for IT and software systems require adaptation before they can be applied effectively in ICS contexts.

2.3 Relevant Standards and Frameworks

Two key standards and frameworks are introduced in this section to provide the necessary context before they are applied in later chapters: the IEC 62443 standard and the MITRE ATT&CK for ICS framework.

2.3.1 ISA/IEC 62443

IEC 62443 is an international standard for the security of Industrial Automation and Control Systems (IACS), developed jointly by the International Society of Automation (ISA) and the International Electrotechnical Commission (IEC). The two organizations publish

identical documents simultaneously: ISA editions are published as ANSI/ISA-62443 in the United States, while IEC editions are published as IEC 62443 internationally.[1] Throughout this thesis, the standard is referred to as IEC 62443, following the international naming convention. The standard is organized into four groups addressing general concepts, policies and procedures, system-level requirements, and component-level technical requirements [22].

IEC 62443-4-2 defines technical security requirements for IACS components, organized across seven fundamental requirement categories: identification and authentication control, use control, system integrity, data confidentiality, restricted data flow, timely response to events, and resource availability [2]. These requirements are graded across four Security Levels (SL), from SL 1, which addresses protection against casual or coincidental violations, to SL 4, which addresses protection against intentional violations using sophisticated means with extensive resources and high motivation [22]. IEC 62443-4-1 defines a secure product development lifecycle and specifies security testing requirements for newly developed IACS components, including penetration testing as one of the security verification and validation testing processes [1].

2.3.2 MITRE ATT&CK for ICS

MITRE ATT&CK for ICS is a publicly available knowledge base of adversarial tactics and techniques observed in real-world attacks on ICS, maintained by the MITRE Corporation [3]. It is structured as a matrix in which each column represents a tactic, which is a high-level attacker objective such as gaining initial access, moving laterally, or causing physical impact. Each column lists the specific techniques that an attacker may use to achieve that objective. The framework is built on the analysis of real-world ICS incidents, which distinguishes it from theoretically derived threat models and makes it a reference grounded in observed attacker behavior.

MITRE ATT&CK for ICS is specifically designed for ICS environments and differs from the enterprise-focused ATT&CK framework in that it includes tactics and techniques that account for the interaction between the cyber and physical domains [3]. For example, it includes tactics such as Inhibit Response Function and Impair Process Control, which have no direct equivalent in the enterprise framework and reflect the physical consequences that attacks on ICS can produce. The framework operates at a level of abstraction that is more specific than high-level threat categories but less specific than individual exploit code, making it applicable across a wide range of ICS environments and attack scenarios.

2.4 Summary

This chapter introduced the core concepts that form the foundation of this thesis. The security challenges specific to ICS were examined from four perspectives: the expanded attack surface from IT and OT integration, the physical consequences of cyber incidents, the conflict between operational availability and security practices, and the long operational lifecycle that makes security by design a fundamental requirement. Threat modeling was introduced as the structured mechanism for analyzing security threats in ICS, with STRIDE and DFDs as the primary threat enumeration technique and system representation method, and IEC 62443 and MITRE ATT&CK for ICS as the key standards and frameworks used throughout this thesis.

[1] For a detailed explanation of the naming conventions and the relationship between the ISA and IEC editions, see the ISA99 Committee FAQ: `https://ruscadasec.com/ISA99-62443-FAQ.pdf`

3 Methodology

This chapter describes the research methodology of this thesis, which adopts DSR as its primary approach. DSR is an established approach in information systems and computer science that generates knowledge through the design, development, and evaluation of artifacts that address real-world problems [20, 35]. This approach is particularly well-suited to this research, where the aim is to create and evaluate practical solutions to identified problems rather than analyze them. The research follows the six-phase DSR model proposed by Peffers et al. [35], which consists of six phases: problem identification and motivation, definition of objectives, design and development, demonstration, evaluation, and communication.

The research is organized as two consecutive DSR cycles, each demonstrating and evaluating a distinct artifact through case study research, together with an exploratory case study that examines how threat modeling can be integrated with an existing cyber resiliency assessment framework and what challenges arise in practice. The two DSR cycles address RQ2 and RQ3, respectively, and the case study addresses RQ4. RQ1 is addressed through the systematic literature review conducted as the first phase of DSR Cycle 1, which establishes the state of the art and identifies the research gaps that motivate the methodology developed in the cycle.

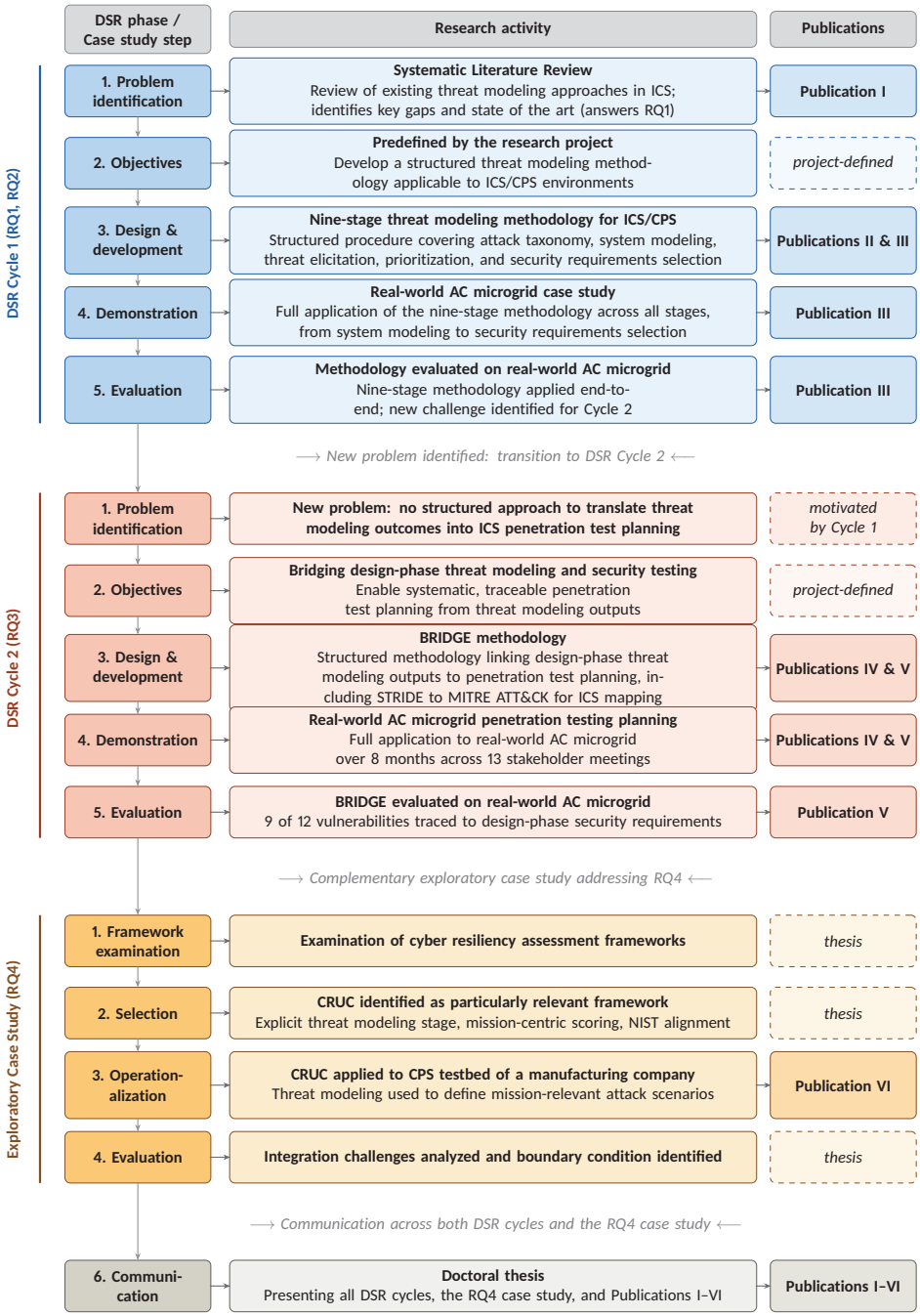
A nine-stage threat modeling methodology developed in Cycle 1 was demonstrated and evaluated as a complete solution to RQ2. A second, distinct problem became apparent once the case study system moved from design to deployment: there was no established way to carry the design-phase outputs forward to guide penetration testing of the newly deployed system. This problem is addressed by RQ3 through the second DSR cycle, which develops and evaluates the BRIDGE methodology as a distinct artifact for connecting design-phase threat modeling to penetration testing. Structuring the work this way keeps the nine-stage methodology as an independently complete and reusable artifact, while allowing BRIDGE to be designed as a general mechanism for extending threat modeling outputs into security testing, rather than as a fixed extension of the nine-stage methodology alone. Table 2 summarizes the mapping of research questions to research methodology and the corresponding publications. Figure 1 illustrates how these studies relate to one another and maps each phase to the corresponding publications.

Table 2: Mapping of research questions to research methodology and publications.

RQ	Research Methodology and Methods	Publications
RQ1	Systematic Literature Review within DSR Cycle 1 Phase 1 [29]	I
RQ2	DSR Cycle 1 with case study research for demonstration and evaluation [35, 41]	II, III
RQ3	DSR Cycle 2 with case study research for demonstration and evaluation [35, 41]	IV, V
RQ4	Exploratory case study research investigating the integration of threat modeling with an existing cyber resiliency assessment framework [41, 57]	VI

3.1 Research Methods

Two main research methods are used in this thesis: systematic literature review and case study research. These methods are applied across the two DSR cycles and the RQ4 case



DSR cycles follow the six-phase model by Peffers et al. (2007). The RQ4 case study uses exploratory case study research following Runeson & Höst (2009) and Yin (2018). Dashed elements indicate no direct publication.

Figure 1: Overview of the research methodology, comprising DSR Cycle 1 (RQ1–RQ2), DSR Cycle 2 (RQ3), and an exploratory case study (RQ4).

study. This research is primarily qualitative in nature, as it focuses on developing and evaluating methodologies through their application to real and representative industrial systems.

The SLR was conducted as the first phase of DSR Cycle 1 to establish the state of the art in threat modeling for ICS and to identify the specific gaps that the research addresses. The SLR followed the guidelines proposed by Kitchenham and Charters [29]. These guidelines were chosen as they provide a structured and reproducible process for searching, selecting, and analyzing literature, which is essential for establishing a credible evidence base. The results of the SLR directly answer RQ1 and are presented in Chapter 4.

Case study research follows the guidelines proposed by Runeson and Höst for software engineering research [41] and the design and methods described by Yin [57]. This thesis is built on practical project requirements, which made case study research the most suitable evaluation approach. Within the two DSR cycles, case study research is used for demonstration and evaluation of the designed artifacts. The primary case study for both DSR cycles is a real-world AC microgrid system developed at Tallinn University of Technology. For RQ4, an exploratory case study was conducted on a CPS testbed of a manufacturing company.

The case studies were conducted as part of real project activities, and the data were collected from these activities. In DSR Cycle 1, the threat modeling was carried out in 13 structured sessions with a team of five participants. The data consisted of the documented outputs of these sessions, such as the DFD, the threat list, the threat priorities, and the selected security requirements [24]. In DSR Cycle 2, the penetration testing process was planned and followed up in 13 technical meetings over eight months. The data consisted of the scoping and planning documents, the penetration testing report, and a follow-up meeting with the penetration tester in which the status of each planned test was documented [26]. Each artifact was evaluated qualitatively by applying it to the real microgrid system. The nine-stage methodology was evaluated by whether each of its stages could be carried out with the information and expertise available in the project. BRIDGE was evaluated by tracing the reported vulnerabilities back to the design-phase security requirements and by comparing the severity rankings produced by CVSS v3.1 and CVSS v4.0.

3.2 DSR Cycle 1: A Structured Threat Modeling Methodology for CPS

The first DSR cycle addresses the gaps identified in the SLR [25]: existing threat modeling approaches for ICS lack structured and repeatable procedures, do not cover the full threat modeling lifecycle, and are rarely validated in real-world deployments. The objective of Cycle 1 was to develop a structured and practical threat modeling methodology that can be applied to ICS and that addresses the identified gaps. This objective was defined after examining existing threat modeling approaches and finding that none of them adequately addressed the identified gaps in ICS threat modeling [25].

The design and development phase produced two related artifacts. The first is a cyber attack taxonomy for microgrid systems [8], developed through a review of the literature on attacks targeting microgrids and smart grids. It serves as a structured knowledge base to support the threat identification stage and was further refined during the case study application [24]. The second artifact is a nine-stage threat modeling methodology for CPS [24], which adapts STRIDE to the specific characteristics of cyber-physical environments. The methodology was developed collaboratively with cybersecurity experts, a software architect, and a power systems expert who acted as the system owner, reflecting the multidisciplinary nature of ICS security work. The methodology was demonstrated through its full application to the real-world microgrid case study [24], covering all nine stages from

attack taxonomy creation through to the selection of security requirements aligned with IEC 62443. The detailed results of this demonstration are presented in Chapter 5.

3.3 DSR Cycle 2: Bridging Threat Modeling and Penetration Testing

The second DSR cycle was motivated by a problem that emerged from Cycle 1: while the threat modeling methodology produced a structured set of identified threats and security requirements, there was no established approach for using these design-phase outputs to plan and guide security testing after the system is deployed. This gap became clear when the microgrid system transitioned from design to deployment and penetration testing was being planned. The objective of Cycle 2 was to develop a methodology that uses threat modeling outcomes to guide ICS penetration testing planning, ensuring that the testing effort is focused on the threats that matter most for the specific system.

The design and development phase produced the BRIDGE methodology, a nine-stage threat modeling-driven penetration testing methodology for newly deployed ICS [27, 26]. BRIDGE uses the outputs of design-phase threat modeling, including identified threats, attack vectors, and DFD components, to structure the scoping and planning of penetration testing. A key part of the methodology is a mapping of STRIDE threat categories to MITRE ATT&CK for ICS techniques. This mapping bridges the high-level abstraction used in design-phase analysis and the mid-level tactical detail provided by MITRE ATT&CK for ICS, which is the appropriate level of specificity for structured test planning. The BRIDGE methodology was demonstrated and evaluated through its application to the same AC microgrid case study, covering the full process from the DFD update through to penetration testing, vulnerability ranking, report analysis, and retesting [26]. The detailed results are presented in Chapter 6.

3.4 Case Study on Threat Modeling and Cyber Resiliency Assessment Integration

Alongside the two DSR cycles, a case study was conducted to address RQ4. The aim of this study was to examine how threat modeling can be integrated with an existing cyber resiliency assessment framework and to identify the practical challenges of operationalizing this integration in a practical CPS evaluation context [30]. CRUC [9] was identified as a particularly relevant framework as it explicitly includes a threat modeling stage, is aligned with the National Institute of Standards and Technology (NIST) Cyber Resiliency Engineering Framework (CREF), and provides a mission-centric scoring mechanism.

CRUC was operationalized on a CPS testbed of a manufacturing company, and an attempt was made to apply the nine-stage threat modeling methodology, developed in DSR Cycle 1, as input to the threat model component of CRUC. During the operationalization, it became clear that CRUC prioritizes threats based on mission impact rather than requiring a comprehensive system-level threat analysis, making a full nine-stage threat modeling exercise disproportionate in this context. The application was therefore discontinued before the threat elicitation stage, and this boundary condition is analyzed in detail in Chapter 7. This study is not a DSR cycle because no new artifact is produced. It is an exploratory case study, following the guidelines of Runeson and Höst [41] and Yin [57], that evaluates how an existing framework applies in a specific context.

3.5 Communication

The outcomes of the two DSR cycles and the RQ4 case study have been communicated through six peer-reviewed publications in international journals and conference proceed-

ings, as listed in the front matter of this thesis. The publications cover the full arc of the research: from the analysis of the state of the art in threat modeling for ICS (Publication I), through the development and demonstration of the threat modeling methodology (Publications II and III), to the development and evaluation of the BRIDGE methodology for threat modeling-driven penetration testing (Publications IV and V), and the investigation of threat modeling integration with an existing cyber resiliency assessment framework (Publication VI). As conference publications are subject to page limits, the analysis of the challenges encountered during the operationalization of CRUC is expanded in Chapter 7 of this thesis.

3.6 Summary

This chapter described the research methodology used in this thesis. DSR was adopted as the primary methodology since the central goal of the research is to design and evaluate practical artifacts. The research is organized as two consecutive DSR cycles alongside an exploratory case study investigating the integration of threat modeling with an existing cyber resiliency assessment framework. The SLR provides the evidence base for the problem identification in Cycle 1 and directly answers RQ1. DSR Cycle 1 developed and evaluated a structured threat modeling methodology for ICS and CPS, addressing RQ2. DSR Cycle 2 developed and evaluated the BRIDGE methodology for threat modeling-driven penetration testing of newly deployed ICS, addressing RQ3. The RQ4 case study investigates how threat modeling can be integrated with the MITRE CRUC methodology, and the challenges encountered during this operationalization are discussed in Chapter 7. Both DSR cycles used a real-world AC microgrid as the primary case study.

4 Threat Modeling in Industrial Control Systems: State of the Art

This chapter presents the results of the SLR conducted as part of DSR Cycle 1 to examine the state of the art in threat modeling for ICS [25]. The review was initiated in the context of a microgrid development project[2], in which the lack of a well-documented and practical ICS threat modeling methodology motivated a systematic examination of the existing literature to identify and evaluate available approaches. The chapter summarizes the review protocol, presents a taxonomy developed to categorize ICS threat modeling methods, analyzes the extent to which the proposed methods cover the full threat modeling process, and identifies the research gaps that motivated the subsequent contributions of this thesis. At the time of its publication, no comparable systematic review of threat modeling methods specifically focused on ICS was identified [25]. The review was designed to serve both academic researchers and practitioners, establishing a shared terminology and taxonomy as a common foundation for discussing ICS threat modeling methods.

4.1 Review Process and Selected Papers

The review was conducted iteratively across three search rounds in May 2021, February 2022, and June 2023, covering five academic databases: IEEE Xplore, Scopus, Web of Science, Science Direct, and ACM Digital Library. Figure 2 summarizes the review protocol [25].



Figure 2: SLR review protocol, including the research questions, search strategy, inclusion criteria, exclusion criteria, and quality criteria [25].

[2]The project was supported by the Estonian Ministry of Education and Research and the European Regional Development Fund (grant 2014-2020.4.01.20-0289).

The cumulative search produced 169 candidate papers, of which 162 were accessible. After applying the selection criteria at the abstract and full-paper reading stages, 36 papers were selected for full analysis, of which 27 were found through the systematic database search and 9 through snowballing. The quality criteria were designed to retain papers that propose or apply a structured threat modeling methodology covering a broad range of threat types, as the review aimed to identify approaches that could be practically followed in a real ICS project rather than methods targeting a single specific attack type.

4.2 A Taxonomy for ICS Threat Modeling

To analyze the selected papers in a structured way, a taxonomy for ICS threat modeling was developed as part of this review [25]. The taxonomy classifies each study across five categories: method type, approach, focus, validation, and application area. Figure 3 shows the structure of the taxonomy.

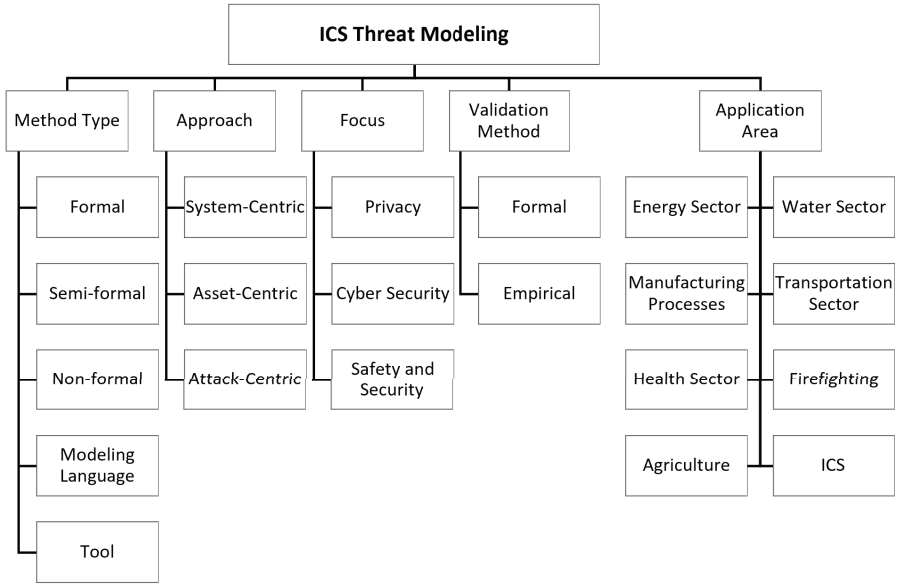


Figure 3: Proposed ICS threat modeling taxonomy used to classify the 36 selected papers [25].

The taxonomy classifies threat modeling methods based on their type, covering non-formal, semi-formal, formal, tool-based, and modeling-language-based methods. Of the 36 selected papers, 19 apply non-formal methods, representing the most prevalent method type in the selected ICS threat modeling literature. Formal approaches are largely absent from the selected papers, as the majority of formal threat modeling methods in the broader literature tend to address one or two attack types, and therefore did not satisfy the quality criteria applied in this review.

The approach category classifies the perspective from which threats are analyzed: asset-centric (starting from valuable assets), attack-centric (starting from possible attacker actions), or system-centric (starting from the system architecture). Of the 36 selected papers, 23 adopt a system-centric approach, making it the most prevalent perspective in the

scope of this SLR. This finding is consistent with the dominance of STRIDE, which analyzes system components against a fixed set of threat categories. Although different authors have classified STRIDE differently, with some treating it as attack-centric [34] and others as system-centric [48], this review treats STRIDE as system-centric because it focuses on system components rather than attacker profiles. The review also identified studies that combine more than one approach, for example by pairing attack-centric with asset-centric or system-centric perspectives.

The focus category classifies whether a study addresses cybersecurity, privacy, safety, or a combination of these. Studies focusing solely on safety were excluded from the review, as analysis focused solely on safety belongs to the engineering domain and addresses hazards rather than cyber threats, placing it outside the scope of this SLR. Of the 36 selected papers, 25 focus solely on cybersecurity, six address both safety and security, two cover both privacy and cybersecurity, two consider all three aspects simultaneously, and one focuses solely on privacy. Given the fundamental importance of safety in ICS environments, where cyber attacks can directly impact physical operations, the fact that eight of the 36 selected papers address safety considerations indicates that this aspect remains insufficiently covered in the ICS threat modeling literature.

The validation category classifies how the proposed threat modeling methods are evaluated. Of the 36 selected papers, 35 use empirical validation, of which 34 apply qualitative methods and one applies a quantitative approach. Qualitative validation is predominantly conducted through use cases and case studies. This may suggest that many of the selected studies represent initial proposals of new methods, where a use case serves as a first demonstration rather than a rigorous evaluation, with quantitative validation potentially following in later publications.

The application area category shows that the energy sector dominates, with 22 papers covering this sector. Other application areas include water, manufacturing, transportation, health, and agriculture, though each of these sectors is represented by a limited number of papers. The dominance of the energy sector is likely due to its critical nature, as many other sectors depend on it, and to the significant damage that cyber attacks on energy infrastructure can cause.

4.3 Threat Modeling Methods and System Mapping Techniques

In this review, STRIDE is classified as a threat enumeration technique rather than a standalone threat modeling method, as it focuses on identifying threats against system components using six security categories without providing a complete procedure covering the full threat modeling process [25]. Nevertheless, STRIDE is the predominant threat enumeration technique across the 36 selected papers, applied in 13 papers. Of these, 11 combine STRIDE with DFD as the system mapping approach, while Ramis Ferrer et al. [36] apply STRIDE using a component table listing system components, trust levels, and entry points, and Rouland et al. [40] apply STRIDE within a formal specification language.

Beyond STRIDE-based analysis, the remaining papers employ a variety of other threat modeling methods. Wuyts et al. [56] apply LINDDUN (Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, Non-compliance), a method that shares a similar approach to STRIDE while focusing on privacy-specific threats, systematically categorizing them against system components using DFD as the system mapping method. Friedberg et al. [17] propose Systems-Theoretic Process Analysis (STPA)-SafeSec, which combines safety analysis and security analysis into a single framework. Lee et al. [31] apply Process for Attack Simulation and Threat Analysis (PASTA), a seven-stage threat modeling framework recommended for CPS threat modeling by

Shevchenko et al. [47]. As PASTA does not itself specify a threat identification procedure, Lee et al. rely on STRIDE for threat analysis and attack trees for attack modeling, and the resulting mitigation strategies remain broad and qualitative rather than tied to individual identified threats. Zografopoulos et al. [58] apply Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) Allegro, which focuses on information asset risk assessment. Amro et al. [6] combine Failure Modes, Effects, and Criticality Analysis (FMECA) with MITRE ATT&CK for ICS to assess cyber risk in a cyber-physical context. Several papers rely on dedicated tools and modeling languages, including securiCAD [55], powerLang [19], and TAMELESS [54], which automate parts of the threat analysis while requiring familiarity with domain-specific modeling languages.

Despite its dominance, the application of STRIDE across the selected papers is not standardized. Papers apply STRIDE in different ways, use different formats to document the elicited threats, and make different choices between STRIDE-per-element and STRIDE-per-interaction without explaining the rationale. There is also no common format for describing the elicited threats, with some papers using structured tables, others linking threats to specific Common Vulnerabilities and Exposures (CVE), and others providing informal descriptions. This lack of a documented and repeatable procedure for applying STRIDE to ICS is one of the key gaps identified by this review [25]. DFD is the most widely used system mapping method, almost always applied alongside STRIDE or LINDDUN. An alternative worth noting is the Colored Petri-Net, used in two papers, which can represent different types of information flows such as data flows, command flows, and power flows simultaneously.

4.4 Coverage of Threat Modeling Stages

To evaluate the coverage of the full threat modeling process across the selected papers, this review identified six threat modeling stages derived from the existing literature, as illustrated in Figure 4 [25].

The analysis of stage coverage across the 36 selected papers reveals varying levels of engagement across the six threat modeling stages, with universal coverage of the core stages and a notable decline in coverage for the stages that require real-world application or rigorous evaluation. Stage 1 (identifying security objectives) is addressed by three papers, representing the lowest coverage among all stages alongside Stage 6. This could be related to the fact that most papers were not applied to real-world systems where organizational policies and regulatory requirements must be considered. Additionally, methods such as STRIDE and LINDDUN might be considered a substitute for this first stage, as they already analyze the system against aspects that could be treated as security and privacy objectives.

All 36 papers cover Stage 2 (system mapping) and Stage 3 (threat enumeration), confirming that these are the fundamental and universally adopted stages of the threat modeling process. This universal coverage reflects the fact that system mapping and threat enumeration are the core activities without which no threat modeling exercise can proceed. Stage 4 (impact and risk analysis) is addressed by 23 papers, while Wuyts et al. [56] explicitly set it out of scope, and the remaining papers do not address it. This indicates that while the majority of the selected papers consider threat prioritization as part of the threat modeling process, a considerable proportion stops at threat enumeration without proceeding to assess the impact or risk associated with the identified threats.

Of the 36 selected papers, 20 address Stage 5 (mitigation), while 12 do not mention it and four explicitly set it out of scope. Among the papers that address mitigation, different approaches to security requirements selection are adopted, with some referencing

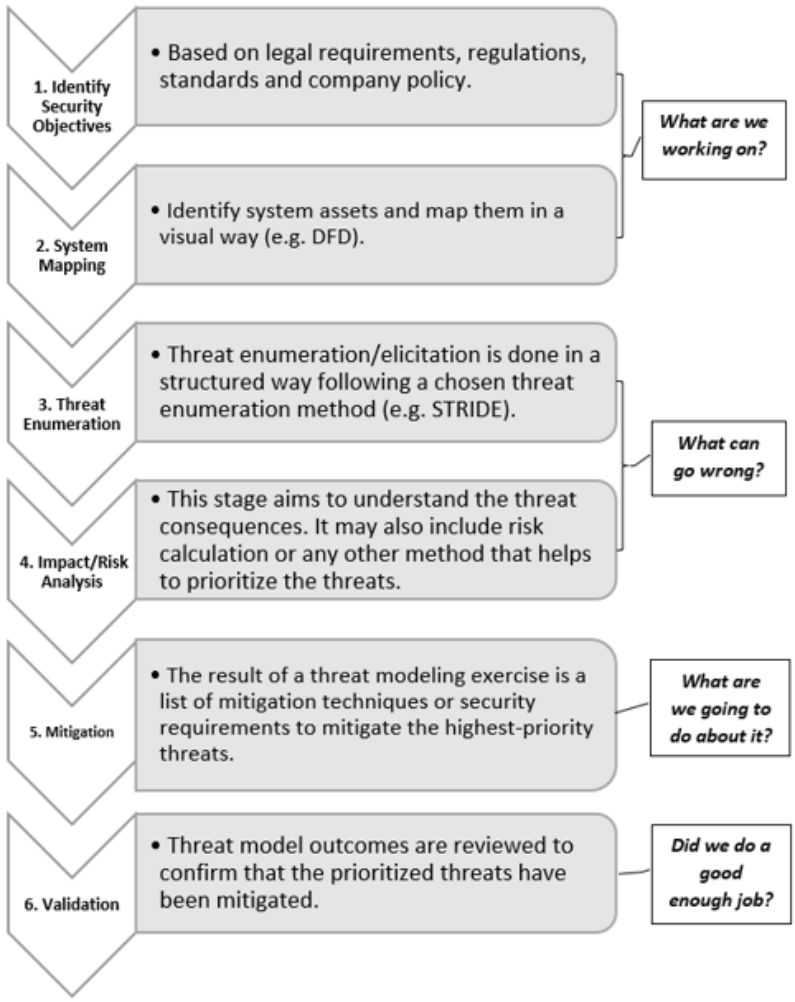


Figure 4: The six threat modeling stages identified through the review, used to evaluate stage coverage across the selected papers [25].

established standards such as IEC 62443 [2] or NIST controls [23], while others propose mitigation techniques without linking them to a recognized framework. A limited number of papers provide a structured approach to security requirements selection by mapping the identified threats to specific standard controls.

The near-complete absence of Stage 6 (validation) is particularly significant, with three papers addressing it. It is important to note that this refers to the validation of the threat modeling outcomes as part of the threat modeling process itself, that is, verifying that the threat model is complete and that the identified threats and security requirements are appropriate. This is distinct from the empirical validation used by the selected papers to demonstrate the applicability of their proposed methods, discussed in Section 4.2. The review identified two types of threat modeling validation that are missing from the literature: validating the threat modeling method itself to confirm that it correctly and completely identifies the threats it is designed to capture, and validating the outcomes of a specific threat modeling exercise to confirm that the most critical threats were elicited

and that suitable security requirements were defined. Both types are largely absent from the selected papers.

4.5 Research Gaps

The SLR identified several research gaps in the existing ICS threat modeling literature [25]. These gaps reflect limitations in procedural coverage, real-world applicability, multidisciplinary communication, validation practices, and the scope of threat modeling application across the Software Development Life Cycle (SDLC). They are summarized below, followed by a discussion of those that directly motivated the contributions of this thesis.

The first two gaps reflect limited procedural maturity in ICS threat modeling. Gap G1 concerns the limited procedural coverage of the threat modeling process: ICS threat modeling is currently more focused on system mapping and threat enumeration and does not provide clear procedures for the remaining stages, while a complete and detailed description of all threat modeling stages is a fundamental requirement for practicing threat modeling in real settings. Gap G1 persists even where broader process frameworks covering the full threat modeling lifecycle exist. PASTA, for example, defines seven stages spanning system modeling to security requirements and mitigation, and is recommended for CPS threat modeling by Shevchenko et al. [47], though without a demonstrated case study. Lee et al. [31], discussed in Section 4.3, later applied PASTA to a generic CPS setup. Neither source provides a worked example detailed enough to guide a real-world project.

Gap G2 concerns the absence of a common standard for applying and documenting STRIDE in ICS environments. While STRIDE is a widely used threat enumeration technique, the selected papers do not use a common language or format to describe the elicited threats. For example, Chen and Huang [12] merge the threat elicitation stage with risk assessment without providing detailed threat descriptions, while Ramis Ferrer et al. [36] link each threat to a specific CVE. This inconsistency makes it difficult to compare results across studies or establish a repeatable practice.

Gaps G3 and G4 concern the real-world applicability of the proposed methods. The selected papers do not discuss the technical and organizational challenges associated with applying threat modeling in ICS environments, nor do they provide guidance on how such challenges can be identified or overcome in practice (G3). This is compounded by the lack of published industrial case studies (G4), which limits the understanding of how threat modeling methods perform under real operational conditions.

Gap G5 concerns the communication aspect of threat modeling. Threat modeling is an interactive activity that requires collaboration between teams with different perspectives and expertise, including system developers, system administrators, security experts, and domain professionals. Compared to software-based systems, the development and maintenance of ICS and CPS require more diverse expertise, ranging from experts in physical processes to system developers. The selected papers do not address how to effectively structure this communication to ensure a common understanding of the identified threats and their implications across multidisciplinary teams, which is a critical requirement for threat modeling to be effective in practice.

Gap G6 concerns the quality and rigor of threat modeling validation. While 35 of the 36 selected papers use empirical validation, this is predominantly conducted through use cases and case studies without clear evaluation criteria. Two papers stand out as exceptions. Suleiman and Svetinovic [50] evaluate the Security Quality Requirements Engineering (SQUARE) method using a case study based on qualitative research, analyzing the method against ten criteria: flexibility, sampling, analyst permissibility, interpretiveness, data sufficiency, coherence, repeatability, completeness, usability, and credibility. Wuyts

et al. [56] evaluate LINDDUN quantitatively against correctness, completeness, productivity, and ease of use. Scandariato et al. [45] applied the same experiment setup to validate STRIDE in the software domain, yet no equivalent structured validation was identified for STRIDE or other widely used threat modeling methods in the ICS context [25]. The second type of validation, namely verifying the outcomes of a specific threat modeling exercise to ensure that the most critical threats were elicited and suitable security requirements were proposed, is largely absent from the selected papers. This represents a clear quality assurance gap in the ICS threat modeling literature, which is consistent with the findings of Tuma et al. for software systems [53].

Gap G7 concerns the scope of threat modeling application across the SDLC. All 36 selected papers apply threat modeling exclusively during the design phase of an ICS, which aligns with the common perception of threat modeling as an early-stage development instrument. However, the knowledge created during threat modeling can be utilized in later SDLC stages, particularly during the security verification and validation testing processes. As ICS and CPS have various types of interfaces, protocols, and data flows, defining a scope and a proper course of action for penetration testing is a significant challenge. Threat modeling can act as an instrument for identifying and prioritizing the system components and information flows to be addressed during penetration testing within given resource constraints, while also enabling knowledge transfer between the early and late stages of the SDLC.

The seven discussed gaps are all addressed to varying degrees by the contributions of this thesis. The lack of a complete and repeatable threat modeling procedure for ICS (Gap G1) motivated the development of the methodology [24] presented in Chapter 5. Gap G2, concerning the inconsistent application and documentation of STRIDE, is addressed through the detailed procedures introduced in Chapter 5 [24]. Gaps G3 and G4, concerning the lack of documented challenges and industrial case studies, motivated the two case studies [24, 26] conducted as part of this research, reported in Chapters 5 and 6. Chapter 7 further contributes to Gap G3 by documenting the practical challenges of using threat modeling as input to a cyber resiliency assessment framework [30].

Gap G5, concerning communication across multidisciplinary teams, is addressed through the detailed threat modeling procedures and shared taxonomy introduced in Chapter 5 [24, 8], which provide a common language for threat elicitation and explicitly identify the stakeholders involved at each stage. Gap G6, concerning the absence of rigorous threat modeling validation, is partially addressed by the contributions of this thesis. While the thesis does not validate the methodology against formally defined criteria, it provides a detailed step-by-step demonstration of the methodology applied to a real-world power system [24]. The outcome validation type is further addressed through the BRIDGE methodology in Chapter 6 [27, 26], which positions penetration testing as a validation mechanism for design-phase threat modeling, providing evidence on whether the selected security requirements were correctly implemented and whether the prioritized threats were effectively mitigated by these requirements. Gap G7, concerning the exclusive use of threat modeling in the design phase, directly motivated the BRIDGE methodology in Chapter 6 [27, 26], which extends threat modeling knowledge to the penetration testing stage of the SDLC. Chapter 7 further examines the use of threat modeling beyond the design phase, as input to cyber resiliency assessment [30]. Table 3 maps each identified gap to the chapter and publications in this thesis that address it.

Saßnick et al. [44], in an independent review published after Publication I [25] of this thesis, examined the use of STRIDE in OT and ICS environments and confirmed the need for a common STRIDE-based methodology for ICS, which aligns directly with Gap G2 identified

Table 3: Mapping of identified research gaps to the chapters and publications that address them.

Gap	Core problem	Chapter	Publications
G1	No complete and repeatable procedure for applying threat modeling in ICS	Ch. 5	Pub. II, III
G2	No common standard for applying and documenting STRIDE in ICS	Ch. 5	Pub. III
G3	Technical and organizational challenges of applying threat modeling not documented	Ch. 5, 6, 7	Pub. III, IV, V, VI
G4	Lack of published industrial case studies	Ch. 5, 6	Pub. III, IV, V
G5	Insufficient attention to communication across multidisciplinary threat modeling teams	Ch. 5	Pub. II, III
G6	Lack of rigorous validation for both threat modeling methods and threat modeling outcomes	Ch. 5, 6	Pub. III, IV, V
G7	Threat modeling applied exclusively in the design phase, not extended to later SDLC stages	Ch. 6, 7	Pub. IV, V, VI

in this review. Saßnick et al. also identified the absence of standardized DFD representations for physical components and the lack of a systematic approach to determining the appropriate level of detail as open challenges in the field. Notably, Saßnick et al. [44] explicitly acknowledged that the methodology developed in this thesis [24] had addressed the physical component representation challenge by introducing distinct notations for physical processes and software-related components, identifying this as a contribution that other approaches in the field had not yet made. This methodology is discussed in detail in Chapter 5.

4.6 Summary

This chapter presented the results of the SLR on threat modeling in ICS, which forms the basis of Publication I of this thesis [25]. The review analyzed 36 papers published between 2012 and 2023 using a proposed taxonomy covering five categories: method type, approach, focus, validation, and application area. STRIDE is the dominant threat enumeration technique in the reviewed papers, typically combined with DFDs; however, its application lacks standardized procedures and consistent documentation practices. The analysis of threat modeling stage coverage showed that all papers cover system mapping and threat enumeration, while coverage drops significantly for the remaining stages, particularly security objective identification and validation. Seven research gaps were identified, reflecting limitations in procedural coverage, STRIDE documentation, the availability of industrial case studies, multidisciplinary communication, threat modeling validation rigor, and the scope of threat modeling application across the SDLC. These gaps are addressed to varying degrees by the contributions presented in the following chapters.

5 Threat Modeling Methodology for Cyber-Physical Systems

This chapter presents the development and application of a nine-stage threat modeling methodology for ICS and CPS environments, which is the main contribution of DSR Cycle 1 described in Chapter 3. The chapter addresses RQ2 and is based on Publications II [8] and III [24]. The chapter first introduces the methodology and the reasoning behind its design choices, then demonstrates how each stage is applied through a real-world AC microgrid case study, presenting the results at each stage. The chapter concludes with a summary of the findings.

5.1 Methodology Overview and Design Rationale

The gaps identified in the SLR presented in Chapter 4 established the need for a threat modeling methodology for ICS that provides detailed and repeatable procedures, covers the full threat modeling lifecycle from system representation to security requirements, captures both cyber and physical dimensions, and has been evaluated in a real-world environment. To address these gaps, a nine-stage threat modeling methodology was developed and evaluated through its application to a real-world AC microgrid system, in the context of an industrial project requiring a structured security assessment at the design phase [24]. Although the methodology was designed to address the identified gap in ICS threat modeling, it may also be applicable to CPS more broadly, owing to the shared characteristics of digital control and physical process interaction. Figure 5 illustrates the proposed methodology, including the iterative feedback loops between stages.

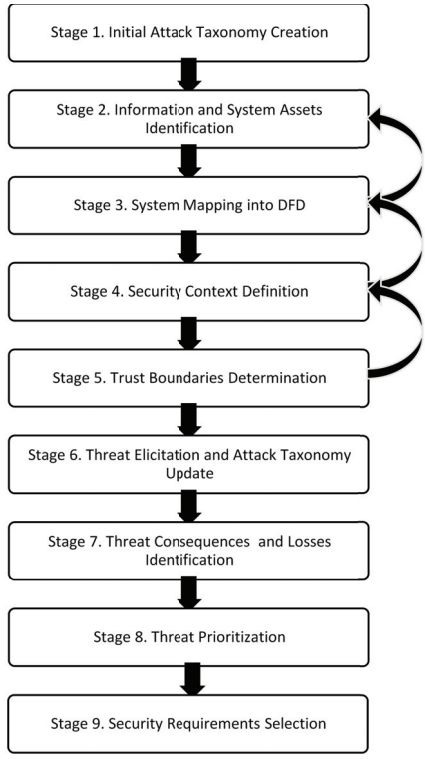


Figure 5: The nine stages of the proposed CPS threat modeling methodology [24].

As illustrated in Figure 5, the methodology consists of nine sequential stages, from initial attack taxonomy creation to security requirements selection. The stages are not strictly linear: Stages 2 and 3 (Information and System Assets Identification and System Mapping into DFD) require iterative refinement and are closely linked in practice, and a similar iterative relationship exists between Stages 4 and 5 (Security Context Definition and Trust Boundaries Determination) [24]. In some cases, the team may also need to revisit earlier stages to refine the asset lists or DFD.

The methodology uses STRIDE-per-element as the threat enumeration technique, integrated with an attack taxonomy as a structured knowledge base and DFD as the system modeling method. The choice of STRIDE-per-element over STRIDE-per-interaction was made on the basis of experimental evidence showing that it produces higher threat completeness [24]. An important design decision in the methodology is the inclusion of physical processes and analog flows in the DFD, even though they do not directly introduce cyber threats. This design choice enables power systems experts to provide domain-specific input during the threat consequence stage, facilitating the assessment of how cyber threats propagate into physical consequences. This is a key distinction between this methodology and standard software-oriented threat modeling approaches that do not account for physical processes.

5.2 AC Microgrid Case Study

The nine-stage methodology was applied to a real-world AC microgrid system developed as part of an industrial project. The application was conducted across 13 structured sessions by a team of five people: three cybersecurity experts, one software architect, and one power systems expert who acted as the system owner. The following subsections present how each stage of the methodology was applied and the key results produced at each step.

5.2.1 Stage 1: Initial Attack Taxonomy Creation

The first stage of the methodology creates an initial attack taxonomy as a structured knowledge base for the system domain. For the microgrid case study, the taxonomy was developed through a review of the literature on attacks targeting microgrids and smart grids, and was published as a standalone contribution [8]. The initial taxonomy classifies attacks into five categories: Reconnaissance Attacks, Device Compromise, Network Compromise, Integrity Attacks, and Denial of Service Attacks.

The taxonomy was treated as a dynamic knowledge base throughout the study [24]. During the threat elicitation process at Stage 6, the taxonomy was updated to reflect new findings. Reconnaissance Attacks were not carried forward as a separate category in the final taxonomy, as reconnaissance activities are inherently represented within the Device Compromise and Network Compromise categories, given that any reconnaissance effort requires the attacker to first gain a foothold in either the device or the network [24]. Under Device Compromise, Escalation of Privilege was added as a new sub-category. Within the Integrity Attacks category, False Data Injection was replaced by two more specific attack types, Forging and Tampering, to better distinguish between the creation of entirely new false data and the modification of existing data. Two device-based availability attacks, Delete/Corrupt Data and System Corruption, were added to the Denial of Service Attacks category, which was renamed Availability Attacks in the final taxonomy [24]. For application within the methodology, the updated taxonomy was reorganized into two functional groups: *Threat Preconditions*, covering Device Compromise and Network Compromise, and *Threat Final Harm*, covering Integrity Attacks and Availability Attacks. Figure 6 illus-

trates the final updated taxonomy as used in the case study application [24].

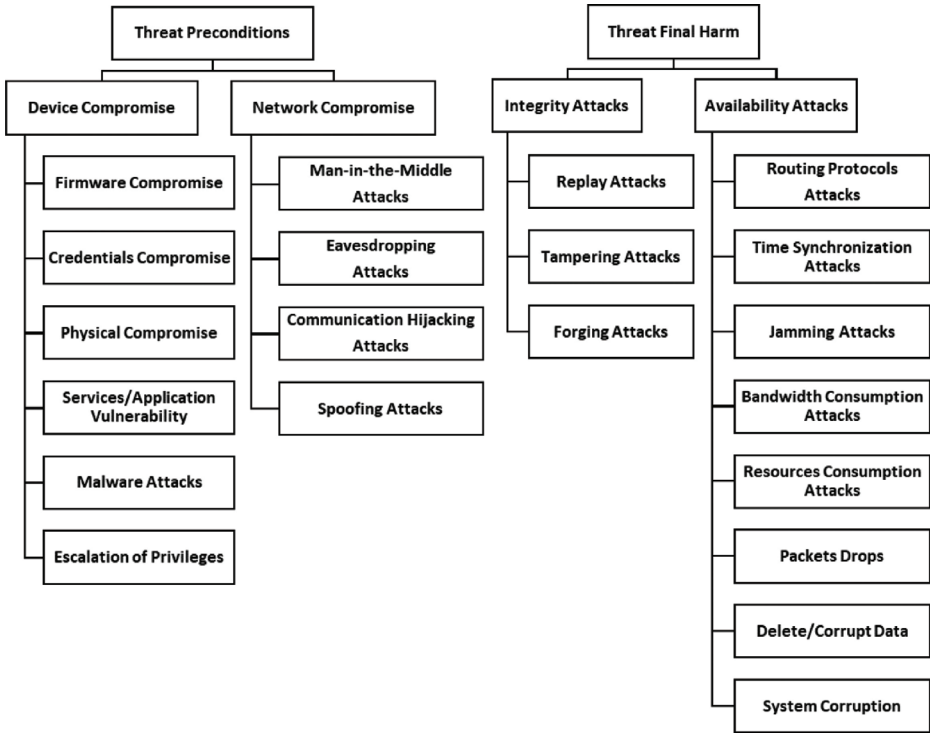


Figure 6: Final version of the attack taxonomy for microgrid systems [24].

5.2.2 Stage 2: Information and System Assets Identification

At the Information and System Assets Identification stage, the team identified all system and information assets in collaboration with the system owner and software architect. The system assets were categorized into six types: software-related processes, measurement device processes, electrical power processes, data stores, external entities, and physical processes. This subcategorization of processes by functional type goes beyond the standard DFD notation, which treats all processes as a single element type, and was introduced to reflect the distinct characteristics of CPS components and to facilitate the threat elicitation and consequence assessment stages [24]. A total of 14 processes, 2 data stores, and 5 external entities were identified. Physical processes were included in the asset identification despite not being subject to STRIDE threat elicitation, as their inclusion in the DFD enables power systems experts to assess how cyber threats propagate into physical consequences during Stage 7 [24].

One notable classification decision at this stage concerned the microgrid admin computer. At the design phase, it was not yet determined whether the device would be organizationally managed or brought in by the admin, so a conservative assumption was made that it could follow a bring-your-own-device model. Based on this assumption, the device was modeled as an external process rather than a regular external entity. This distinction is important in the context of STRIDE-per-element: external entities attract only Spoofing and Repudiation threats, whereas processes attract all six STRIDE categories. Since the organization can still enforce security policies on a bring-your-own-device, modeling it as

an external process enables a full set of threats to be elicited while reflecting the device's untrusted status [24]. However, as discussed in Chapter 6, this representation was revised in the updated DFD once it was established that the admin computer was organizationally managed, allowing it to be reclassified as a regular internal process and treated as any other system component with a full set of security requirements applied to it [26].

Table 4 shows the five information asset categories and their sub-categories, identified through collaborative discussions among the team members. Two of the five categories, control data and operational data, are time-sensitive, which is a characteristic typical of ICS and CPS environments and has direct implications for which security requirements are feasible to derive, as real-time constraints may restrict the applicability of certain requirements [24]. The remaining three categories, metering data, produced data, and device management data, are not time-sensitive and are therefore subject to a broader range of applicable security requirements.

Table 4: Information asset categories identified in the AC microgrid system [24].

Category	**Sub-category**	**Category Description**
Control Data		Control commands
Operational Data	Measurement Data Alarm Data Event Data	Data representing the operational status of assets
Metering Data		Energy meter data for billing
Produced Data	Prediction Data External Data	Processed and external data, e.g., weather forecasts and electricity prices
Device Management Data	Service Data System Logs Configuration Data	Administrative and maintenance data, including service access logs and device configurations

5.2.3 Stage 3: System Mapping into DFD

The DFD was created using the identified system and information assets as input. As an intermediate step, a Data Flow Mapping Table was created to map each data flow to its relevant information asset types and document the direction of network connection initiation, which informed both the DFD structure and the threat elicitation stage [24]. To make the DFD more suitable for a CPS context, three enhancements were made to the standard DFD elements [24].

Physical processes were added as a distinct element type and placed outside the trust boundaries to make clear that they are not directly subject to cyber threats, although they are affected by them indirectly. A color-coding scheme was introduced to differentiate processes by function: blue for software-related processes, yellow for measurement device processes, and gray for electrical power processes. A red circle symbol was added to indicate aggregation points where multiple devices of the same type connect through a single representative node. These enhancements provide better visibility of the system and make it easier for power systems experts to follow the DFD and contribute to Stage 7, Threat Consequences and Losses Identification. Figure 7 illustrates the DFD produced at this stage, which captures the full microgrid architecture and all data flows between components. The trust zones shown in the figure were subsequently defined at Stages 4 and 5

and are included here for reference.

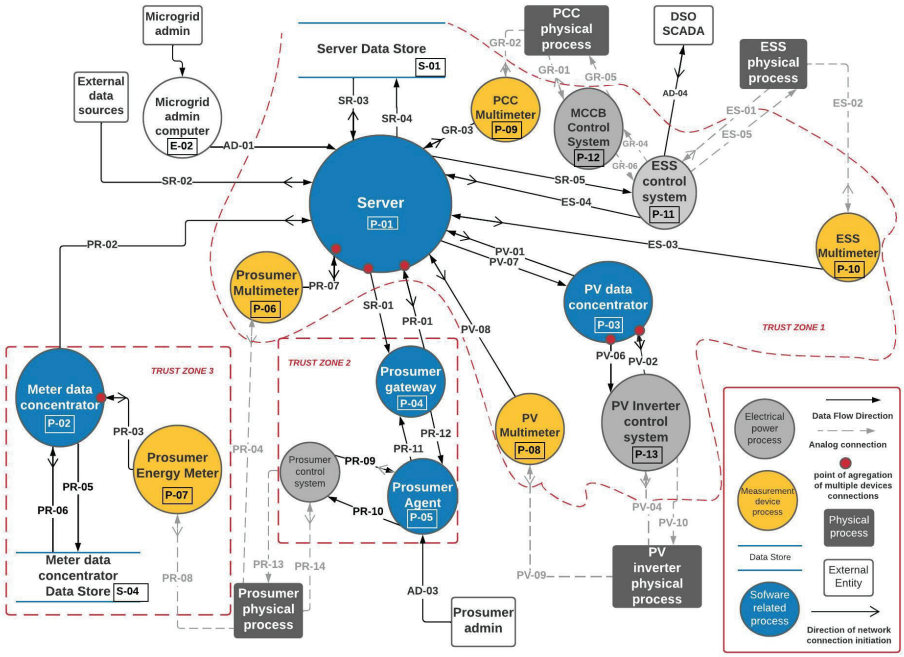


Figure 7: Data Flow Diagram of the AC microgrid system with trust zones defined at Stages 4 and 5 [24].

5.2.4 Stages 4 and 5: Security Context Definition and Trust Boundaries Determination

The security context definition required the team to establish the key assumptions about the deployment environment, the trusted and untrusted entities, and the scope of the threat modeling exercise. Several physically secure assets were identified, including the microgrid server, Energy Storage System (ESS) and Photovoltaic (PV) components, and Moulded Case Circuit Breaker (MCCB) and ESS control systems, whose physical locations were considered secure against unauthorized physical access. This assumption meant that physical compromise was excluded as a precondition attack vector for these assets during the threat elicitation stage.

Four operator groups were identified as internal actors: microgrid admin, Distribution System Operator (DSO) admin, prosumer admin, and maintenance admins (third-party vendors). The first two were treated as trusted entities in terms of intent; the latter two were considered partially trusted, as they may pursue unauthorized goals. It was acknowledged that the devices of all operator groups may be compromised or their credentials stolen regardless of their trust level, meaning that credential-based attacks and device compromise remained within scope for all groups. Regarding external threat actors, nation-state actors were excluded from scope, as compromising a microgrid system was judged unlikely to provide significant political, economic, or military advantage. Terrorist groups were similarly excluded, as their objective is typically to create widespread societal impact through visually demonstrable physical harm, which an attack on a single industrial microgrid system would be unlikely to achieve. Cyber criminals, hacktivists, and malicious prosumers were retained as the relevant threat actors.

In terms of attack types, supply chain attacks were excluded from scope, as they involve hardware-based modifications introduced at the manufacturing stage, which are outside the boundaries of a design-phase threat modeling exercise. Physical attacks such as equipment theft were excluded because the study focused on threats executed through cyber means. It is important to note, however, that this exclusion does not apply to cyber attacks that cause physical consequences, as such threats remained within scope and constitute one of the key characteristics of the methodology. The system owner confirmed that the system operates over a wired communication network, which led to the exclusion of jamming and packet drop attacks, as these are specific to wireless communication environments. Privacy threats were also excluded, as the system does not generate, store, or process sensitive personal data.

Based on the security context established at Stage 4, three trust zones were defined at Stage 5 and added to the DFD produced at Stage 3. Trust Zone 1 covers the physically secure components of the microgrid site, grouped together based on their shared physical security characteristics and the wired communication network of the site. Trust Zone 2 covers the prosumer-owned components. Trust Zone 3 covers the metering subsystem, including the meter data concentrator, its data store, and the prosumer energy meter. One of the most sensitive decisions during the trust boundary discussion was the assumption that the prosumer energy meter is sealed and its network is trusted, which resulted in its inclusion in Trust Zone 3. This assumption led to the exclusion of a significant number of threats related to prosumer energy meter security, such as false data injection attacks that a malicious prosumer could launch to manipulate billing data.

5.2.5 Stage 6: Threat Elicitation and Attack Taxonomy Update

The threat elicitation stage applied STRIDE-per-element to all processes, data stores, external entities, and the external process in the DFD. For data flows, only the seven flows crossing trust boundaries were analyzed, as data flows within the same trust zone are considered trusted and were therefore excluded from threat analysis. Physical processes were not subject to STRIDE elicitation, as they are not directly targeted by cyber threats; however, they are included in the DFD for consequence assessment at Stage 7.

To ensure systematic and repeatable application, the methodology introduced a structured elicitation procedure: for each DFD element, the team identifies the relevant information assets, selects a STRIDE threat category, consults the attack taxonomy for applicable attack vectors, and produces a structured threat definition. Each threat definition comprises three components: a precondition attack state from the taxonomy (device or network compromise), one or more applicable attack vectors, and the impacted information assets with their state (in use, in transit, or at rest). The elicitation produced a total of 110 threats distributed across the six STRIDE categories, as shown in Table 5, covering 7 data flows, 13 processes, 2 data stores, 4 external entities, and one external process.

5.2.6 Stage 7: Threat Consequences and Losses Identification

For each of the 110 elicited threats, the team identified the potential threat consequences and the associated losses. This stage was led by the power systems expert, as assessing the physical consequences of cyber threats in an ICS environment requires domain-specific knowledge that cybersecurity experts alone cannot provide. Thirteen threat consequence definitions were identified, covering impacts such as overloading or exceeding operational limits, disconnection from the microgrid, interruption of energy transfer, misleading safety operations, and leakage of information for further attacks. Each consequence definition was linked to one or more of ten loss definitions, which were assigned priority levels by

Table 5: Number of threats elicited per STRIDE category [24].

STRIDE Category	Number of Threats
Spoofing	12
Tampering	23
Repudiation	17
Information Disclosure	22
Denial of Service	26
Elevation of Privilege	10
Total	**110**

the system owner. The highest-priority losses, rated Critical or High, include loss of life or injury to people, loss or damage to property, and loss of power generation, reflecting the safety-critical nature of the microgrid environment. The full consequence and loss definitions are provided in Publication III [24].

5.2.7 Stage 8: Threat Prioritization

Addressing all 110 identified threats was not feasible within the scope of the project due to time and resource limitations, so a threat prioritization procedure was applied to focus on the most critical ones. As detailed system information such as operating systems, protocols, and product vendors was not yet available at the design phase, the team did not have solid criteria for calculating the likelihood of the identified threats. Beyond this practical limitation, likelihood-based assessment can present significant challenges in ICS environments. Likelihood estimates are difficult to quantify reliably and may be subject to expert bias, potentially leading to the underestimation of critical threats [24]. Moreover, in safety-critical environments such as ICS, where certain threats could lead to loss of life or physical harm, an impact-driven approach may be more appropriate, as it ensures that the severity of potential consequences remains the primary basis for prioritization decisions. An impact assessment was therefore performed instead of a full risk assessment, tracing the loss priorities through threat consequences back to individual threats.

For each threat, the cumulative count of losses at each priority level (Critical, High, Medium, and Low) was computed by summing the priority counts of all threat consequences associated with that threat. Threats were then ranked using a sequential sort, first by the count of Critical-level losses, then High, then Medium, and then Low. This approach was deliberately chosen over an alternative method that replaces each priority level with a numeric weight, as an aggregated numeric score would not preserve the count of Critical-level consequences, which is essential information when prioritizing threats in ICS environments where certain impacts must be prevented regardless of their likelihood.

5.2.8 Stage 9: Security Requirements Selection

At the final stage, the cybersecurity team selected security requirements for the prioritized threats, using IEC 62443-4-2 [2] as the reference standard. Requirements were selected at Security Levels 1, 2, and 3, with SL 4 excluded from scope, as its requirements address threats requiring an advanced level of sophistication and extensive resources, a profile consistent with state-sponsored actors, which were excluded from the threat actor scope at Stage 4. As an example, for a critical spoofing threat targeting the server, in which a threat agent masquerades as the server, the requirements selected included Component Requirement (CR) 1.2 (software process and device identification and authentication), CR 1.2 Requirement Enhancement (RE)(1) (unique identification and authentica-

tion), and CR 6.2 (continuous monitoring). The selected requirements were reviewed by the full team to confirm their feasibility in the context of the system design.

5.3 Key Findings

The inclusion of physical processes in the DFD proved to be a valuable design decision. Although physical processes do not introduce cyber threats, representing them in the DFD created a shared system representation that both cybersecurity and power systems experts could work with. The power systems expert could trace physical consequences directly from the DFD, while the cybersecurity team could follow how cyber threats propagate into the physical domain, creating a common language between the two groups. In multidisciplinary teams, the system representation plays a key role in supporting effective collaboration, and the DFD enhancements introduced in this methodology were designed with that requirement in mind.

The attack taxonomy served as an effective knowledge base during the elicitation stage. It helped structure the threat definitions consistently and reduced reliance on individual expert judgment. The cross-checking of elicited threats against published reference documents also identified a limited number of additional threat types that the initial taxonomy had not included, confirming the value of treating the taxonomy as a dynamic knowledge base throughout the study.

The impact-based prioritization procedure provided a clear and traceable way to rank threats without requiring information that is typically unavailable at the design phase, such as software versions, operating system details, and vendor specifications. The procedure ranks threats by sequentially sorting the cumulative count of losses at each priority level, starting from Critical down to Low. This approach preserves the severity hierarchy of consequences and avoids the distortion that can occur when numeric weights are aggregated, whereby a large number of low-severity impacts could outweigh a smaller number of critical ones.

The connection to IEC 62443 at the final stage provided a direct link between the threat modeling outputs and a recognized industrial standard for ICS security. This link helped ensure that the security requirements produced by the methodology are actionable and aligned with established engineering practice, rather than remaining at an abstract level. The ability to derive IEC 62443 component requirements directly from the prioritized threat list was identified as one of the key practical contributions of the methodology.

The methodology has also inspired related work in the power engineering domain, where it was adopted as a starting point for a Direct Current (DC) microgrid threat modeling study led by researchers from the power engineering domain, to which the author contributed as a co-author [4]. That study applied the nine-stage structure to a DC microgrid testbed, adapting the system representation and asset identification stages to reflect the characteristics of a DC power architecture. The adoption of the methodology by researchers from outside the cybersecurity domain suggests that the documented, step-by-step structure is accessible to practitioners with different backgrounds, and that the approach can serve as a reference point for threat modeling in CPS contexts beyond the system on which it was originally developed. This is consistent with the observation in the SLR that the lack of documented procedures is one of the key barriers to wider adoption of threat modeling in ICS practice, and that structured methodologies with real-world demonstrations may have greater potential for uptake than theoretical frameworks alone [25].

5.4 Summary

This chapter presented the nine-stage threat modeling methodology for ICS and CPS, developed in the context of a real industrial project and in response to the research gaps identified in Chapter 4, and demonstrated its application through a real-world AC microgrid case study. The application produced 110 structured threats across all six STRIDE categories, 13 threat consequence definitions mapped to 10 prioritized loss types, and a set of IEC 62443 component requirements selected at Security Levels 1, 2, and 3 for the highest-priority threats [24]. The methodology's key design decisions include the enhancement of the DFD notation to represent physical processes, enabling cyber-to-physical consequence tracing by multidisciplinary teams, and the use of impact-based prioritization that avoids likelihood estimates, making the methodology applicable at the design phase before implementation details are known. The direct link to IEC 62443 ensures that the outputs are actionable in a real engineering context rather than remaining at an abstract level.

6 From Design to Security Testing: Threat Modeling-Driven Penetration Testing of Newly Deployed ICS

This chapter presents the BRIDGE methodology and its application to the microgrid case study introduced in Chapter 5. BRIDGE is the main contribution of DSR Cycle 2 described in Chapter 3. It addresses the research Gap G7 identified in Chapter 4, which concerns the lack of a structured approach for extending design-phase threat modeling outputs to later stages of the system development lifecycle. The chapter is based on Publication V [26], which substantially extends a preliminary conference version published as Publication IV [27] by formalizing the proposed approach as the nine-stage BRIDGE methodology, and presenting a full case study application with detailed findings and validation against design-phase outputs. The chapter discusses the motivation for BRIDGE and what distinguishes it from existing approaches, describes each stage of the methodology, and presents the results and key findings from the microgrid case study.

6.1 Motivation and Design Rationale

The nine-stage threat modeling methodology presented in Chapter 5 produces a structured set of outputs: a system DFD, a list of 110 threats enumerated and categorized using STRIDE, threat consequence and loss definitions, threat priorities, and security requirements derived from IEC 62443 for the prioritized threats. These outputs capture detailed knowledge about the system that was costly to produce and represent the security understanding established during the design phase. In common practice, however, this knowledge is often not carried forward into the security testing phase. The scope and test cases are typically defined without reference to the design-phase analysis, leaving the penetration tester without the context needed to target the most critical threats [26].

IEC 62443 defines four distinct verification and validation testing processes for a newly deployed system: security requirements testing, threat mitigation testing, vulnerability testing, and penetration testing [1]. While the standard does not prescribe a sequence for these processes, it would be logical to conduct security requirements testing and threat mitigation testing before penetration testing, as this helps ensure that defined requirements are verified and that identified threats have been addressed before an adversarial test is carried out. However, in small and medium-sized ICS projects with limited resources, stakeholders may decide to prioritize penetration testing and exclude the earlier testing processes. In such cases, penetration testing becomes the only process through which the implementation of design-phase security requirements is assessed, a role it was not primarily intended to fulfill. This is precisely what occurred in the microgrid project examined in this chapter, where resource and time constraints led the stakeholders to prioritize penetration testing within the IEC 62443 verification and validation processes, excluding security requirements testing and threat mitigation testing entirely.

BRIDGE is designed to address this situation by providing a structured mechanism for carrying design-phase threat modeling outputs forward into penetration testing. Its final stage also analyzes the penetration testing findings against the design-phase security requirements, which is particularly important when the earlier verification and validation processes were not completed. A key conceptual distinction in BRIDGE is the use of the term *threat modeling-driven* rather than *threat modeling-based* penetration testing. Existing threat modeling-based approaches create a threat model during the testing phase to support test planning [26]. For example, Salzillo et al. and Süren et al. included threat modeling as one of the stages of the penetration testing process for Internet of Things (IoT) systems. Salzillo et al. identified the applicable threats from a threat catalog and

mapped them to attack patterns to plan the tests [42], whereas Süren et al. decomposed the system into attack surfaces and relied on a list of IoT security weaknesses compiled from the OWASP IoT Top 10 [51]. In the ICS domain, Rencelj Ling et al. followed the alternative sequence, in which the results of a penetration test of a remote terminal unit were used to build a threat model in the Meta Attack Language and to generate attack graphs [37]. BRIDGE, by contrast, uses the comprehensive threat model produced during the design phase as the primary input to the testing process. This distinction has an important practical consequence: the relationship between threat modeling and penetration testing becomes bidirectional. In one direction, the design-phase threat model guides the scoping and planning of the penetration tests. In the other direction, the penetration testing results are used to verify whether the specified security requirements were correctly implemented and to validate whether the prioritized threats were effectively mitigated by these requirements. This bidirectionality makes the threat model a living artifact rather than a one-time design document.

Gap G7 was identified through the SLR, whose search terms focused on threat modeling for ICS and did not include penetration testing terms [25]. To complement the SLR from a penetration testing perspective, a structured search was conducted in November 2026 across the five databases used in Publication I. The search combined terms for threat modeling and penetration testing with terms for the application domain, using both full terms and abbreviations, such as “industrial control system” and ICS, “operational technology” and OT, and “cyber-physical system” and CPS, together with SCADA, STRIDE, and the spelling and plural variants of each term. The search returned 41 records, 21 after removing duplicates. Among the records published before the BRIDGE case study was completed in September 2023, two studies combined threat modeling and penetration testing [42, 37], and neither used design-phase threat modeling outputs to plan the penetration testing of the system, as discussed above. Backward citation searching identified one further study outside the ICS domain, in which the threats and security controls identified through a design-phase risk analysis of a cloud application guided penetration testing in the operational phase [11]. This limited number of earlier studies was not sufficient to support a systematic literature review on this topic at the time the BRIDGE methodology was developed.

Related studies published after the BRIDGE case study include work on automated threat modeling and penetration test planning for ICS [18] and the generation of penetration test cases from software architecture models [43], which indicates growing research interest in connecting threat modeling and penetration testing. In these studies, the threat model is built as part of the testing approach itself, for example from a threat catalog or an architecture model, rather than taken from the design-phase threat modeling of the system under test. The gap addressed by BRIDGE was therefore not addressed in the studies identified, either before or after the case study was conducted.

6.2 The BRIDGE Methodology

BRIDGE is a nine-stage threat modeling-driven penetration testing methodology for newly deployed ICS [26]. Figure 8 illustrates the methodology, structured across three lifecycle phases: design, implementation, and testing.

Stage 1, *ICS Threat Modeling*, is conducted during the design phase and corresponds directly to the nine-stage methodology described in Chapter 5. This stage produces the system DFD, the threat list, the consequence and loss definitions, the threat priorities, and the security requirements. Unlike in threat modeling-based penetration testing approaches, this stage is initiated before the system is deployed rather than at the start of

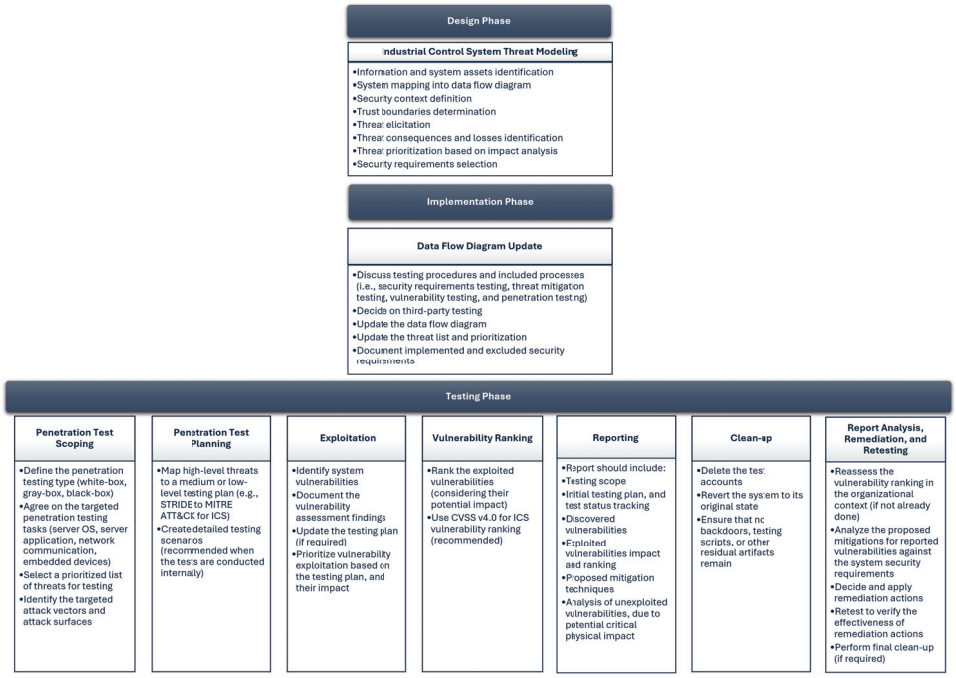


Figure 8: The BRIDGE methodology for threat modeling-driven penetration testing of newly deployed ICS [26].

the testing process. The resulting threat model forms the foundation for all subsequent stages.

Stage 2, *Data Flow Diagram Update*, begins during the implementation phase. It involves discussions about the system testing procedures, covering whether component, integration, and system tests will be conducted separately, and which security verification and validation processes will be applied to the newly developed system components. A decision about whether testing will be conducted internally or by a third party should be made as early as possible to allow sufficient time for external arrangements if required. The DFD is updated at the conclusion of the implementation phase to reflect the current state of the system before testing begins. The cybersecurity team also revises the threat list at this stage, including the threat prioritization. It is further recommended to document which security requirements have been implemented and which have been excluded, as this information supports test selection and follow-up during the report analysis stage.

Stage 3, *Penetration Test Scoping*, defines the boundaries and objectives of the penetration test based on the updated threat model. During this stage, stakeholders agree on the testing type (white-box, gray-box, or black-box) based on the testing goals. This includes selecting the penetration testing task categories, which define the type of system components and attack surfaces to be tested, such as server operating system, server application, network communication, and embedded device testing [46], identifying the targeted attack vectors, selecting a prioritized subset of threats for testing, and defining what is explicitly excluded from the testing scope. The scoping stage produces structured

documentation that guides the tester throughout the testing phase.

Stage 4, *Penetration Test Planning*, translates the scoping outputs into a concrete testing plan. A key contribution at this stage is a mapping of STRIDE threat categories to MITRE ATT&CK for ICS techniques. STRIDE, used during the design phase, operates at a high level of abstraction suitable for early-stage analysis. MITRE ATT&CK for ICS provides a mid-level taxonomy that includes details about attack tactics and techniques, which is the appropriate level for test planning. The mapping bridges these two levels of abstraction, enabling the design-phase threat analysis to be systematically carried forward into test planning. When testing is conducted internally, the planning stage may proceed to a lower level of detail, documenting specific testing procedures and scenarios for each selected technique. When testing is outsourced to a third party, as in the microgrid case study, the plan typically remains at a higher level, identifying which ATT&CK techniques should be tested and providing the rationale for each selection, while leaving the detailed technical procedures to the tester.

Stages 5 through 8, *Exploitation*, *Vulnerability Ranking*, *Reporting*, and *Clean-up*, cover the execution and documentation of the penetration testing activities. These stages can be executed sequentially or in parallel, depending on the project's complexity and available resources. For vulnerability ranking, the methodology recommends Common Vulnerability Scoring System (CVSS) v4.0 rather than CVSS v3.1, because CVSS v4.0 explicitly models the impact on subsequent systems and safety-related consequences, both of which are relevant in ICS environments. The clean-up stage ensures that all testing accounts and artifacts are removed and system configurations are restored before the system becomes operational.

Stage 9, *Report Analysis*, *Remediation*, *and Retesting*, concludes the methodology. At this stage, the penetration testing findings are analyzed against the security requirements defined during the design phase, verifying that the most critical requirements were implemented and identifying any gaps. This analysis is particularly important when penetration testing was prioritized over security requirements testing and threat mitigation testing during the verification and validation process. Remediation actions are defined, applied, and verified through retesting, with a final clean-up conducted if required after retesting.

6.3 Microgrid Case Study

The BRIDGE methodology was applied to the same microgrid system introduced in Chapter 5, covering the full process from the initial DFD update through to report analysis and retesting. The following subsections present the key decisions and results at each stage.

6.3.1 DFD Update and Scoping

The penetration testing process was conducted over eight months, from February to September 2023, involving five stakeholders across 13 technical meetings. The stakeholders included the cybersecurity team, a power systems expert, a software architect, and a third-party penetration tester. The decision to engage a third-party tester was made during the implementation phase to ensure an independent perspective on the system's security posture and to make effective use of the limited testing time available within the project.

The DFD update stage revealed several design changes that had occurred since the initial threat model was produced. These changes had direct consequences for the threat list, with a number of previously identified threats suppressed as irrelevant and new threats added to reflect the updated system configuration. The update also confirmed that the administrator workstation was managed by the same organization operating the

microgrid and was therefore represented as an internal process within a separate trust zone, reflecting its distinct security posture and access pattern.

The scoping stage established gray-box testing as the agreed testing type, with the DFD, high-level network diagrams, and the prioritized threat list provided to the tester. Source code was not disclosed. The scope focused on server operating system, server application, and network communication testing. Embedded device testing was deprioritized due to time and resource constraints and because the embedded components were supplied by established ICS vendors that apply baseline security controls. This decision was explicitly justified by the consequence analysis from the design phase, which identified the in-house software platform, named Server in the DFD, as the component with the most critical safety implications. The scoping stage also defined the rules of engagement, including the exclusion of social engineering attacks, the handling of denial-of-service testing, and the agreement on remote testing without physical on-site access.

6.3.2 Test Planning: STRIDE to MITRE ATT&CK for ICS Mapping

The test planning stage produced a structured testing plan based on a mapping of STRIDE categories to MITRE ATT&CK for ICS techniques. Three attack precondition tactics were defined: Initial Access, Discovery, and Lateral Movement. For each STRIDE category (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege), the applicable ATT&CK for ICS techniques were identified and documented. This produced a testing plan in which the selected ATT&CK techniques were traceable to the STRIDE categories identified during the design phase, with stakeholders agreeing on which techniques to include based on the agreed scope and available resources.

The mapping required interpretive decisions rather than a mechanical one-to-one correspondence. Some techniques appear under multiple STRIDE categories, reflecting the reality that a single attack may serve multiple threat objectives. For example, the adversary-in-the-middle technique was categorized under information disclosure, tampering, spoofing, and denial of service, because an attacker intercepting traffic between two nodes could collect sensitive data, modify packets, inject false data, or block traffic from reaching its destination.

6.3.3 Vulnerability Ranking: CVSS v3.1 vs. CVSS v4.0

Twelve software vulnerabilities were reported by the penetration tester. The tester initially scored them using CVSS v3.1, producing a distribution of 3 Critical, 2 High, 3 Medium, and 4 Low severity vulnerabilities. The same vulnerabilities were subsequently evaluated using CVSS v4.0, considering the threat, environmental, and supplemental metrics, which allowed the security requirements and the safety-related consequences of the system to be reflected in the severity assessment [16]. Figure 9 shows the resulting severity distribution under both scoring systems.

Under CVSS v4.0, the distribution shifted to 4 Critical, 2 High, 2 Medium, and 4 Low. The increase in Critical-severity findings was driven mainly by the modeling of subsequent system impact and safety consequences in CVSS v4.0. For example, a denial-of-service vulnerability that scored 7.4 (High) under CVSS v3.1 increased to 8.3 (High) under the CVSS v4.0 base metrics, as its impact on subsequent systems was included, and to 9.2 (Critical) once the safety consequences identified during the design phase were added through the environmental metrics. Conversely, a grouped vulnerability that received a high score under CVSS v3.1 was slightly reduced from 8.8 to 8.6 under CVSS v4.0, because the confidentiality requirement for this system had been established as low during

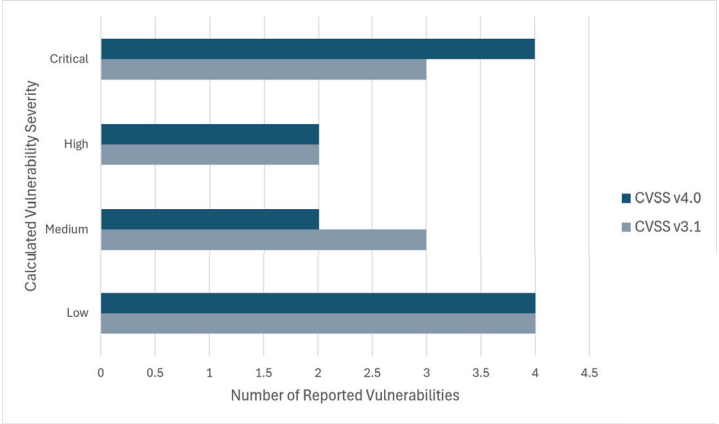


Figure 9: Comparison of vulnerability severity ratings under CVSS v3.1 and CVSS v4.0 [26].

the design phase, and the vulnerability did not introduce safety-relevant consequences. These examples show that, in this case study, CVSS v4.0 produced severity rankings better aligned with the system context than CVSS v3.1. The full CVSS v3.1 and CVSS v4.0 metric values for these examples are reported in Publication V [26].

6.3.4 Validation Against Design-Phase Outputs

The report analysis stage produced the most significant finding of the case study. The proposed mitigations for each of the reported vulnerabilities were analyzed against the security requirements selected during the design phase. The analysis confirmed that mitigation techniques for 9 of the 12 reported vulnerabilities had already been defined through the selected security requirements. In these cases, the vulnerabilities existed because the requirements had not been fully implemented or had been incorrectly enforced, rather than because they were not anticipated during design.

As a concrete example, one of the critical findings was an SQL injection vulnerability that allowed the penetration tester to extract credentials from a system component and then mount a spoofing attack. The spoofing consequences, including overloading operational limits, misleading safety operations, and limited energy availability, had all been identified during the design-phase threat elicitation. The relevant security requirement, IEC 62443-4-2 CR 3.5 (Input Validation), had been selected during design; however, it had not been correctly implemented. This example illustrates how BRIDGE closes the loop between the design and testing phases: a requirement defined during design, linked to an identified threat, was later confirmed as unimplemented through penetration testing.

6.4 Key Findings

The application of BRIDGE to the microgrid system produced several findings relevant to ICS penetration testing practice more broadly. The case study covered the full methodology from the initial DFD update through to report analysis and retesting, making it possible to evaluate both the individual stages and the overall approach under real project conditions. These findings are discussed in turn below, with emphasis on their implications beyond this specific system and project context.

The bidirectional relationship between threat modeling and penetration testing proved to be practically achievable in this ICS project. The design-phase threat model

provided concrete and justified inputs to the scoping and planning stages, and the testing results provided feedback on whether security requirements had been correctly implemented. However, this validation is inherently partial: a successful penetration test can confirm that a requirement was not implemented or was incorrectly enforced, whereas failed tests provide supporting evidence that requirements were correctly implemented. In the context of threat modeling-driven penetration testing, it is therefore recommended that the penetration testing report document not only successful tests but also failed tests and tests that were excluded due to time limitations or unfulfilled preconditions. Reviewing the testing plan status in this level of detail allows stakeholders to identify which threats from the prioritized list were not tested and to assess the completeness of the validation against the design-phase security requirements.

The STRIDE-to-ATT&CK mapping proved to be a productive intermediate step that bridges the abstraction gap between high-level design analysis and executable test planning. The mapping is interpretive rather than mechanical, because both STRIDE and MITRE ATT&CK for ICS are qualitative, expert-driven frameworks that do not share a common formal structure. The mapping documented in this study, including the rationale for overlaps and exclusions, provides a reusable foundation that future ICS penetration testing projects can adopt directly or extend to cover additional ATT&CK matrices, such as the Enterprise or Mobile matrices, where IT components are present in the system.

Vulnerability ranking is a critical step in any penetration testing engagement, as it determines the order in which remediation efforts are prioritized. In ICS environments, standard vulnerability scoring systems may not adequately capture the safety and operational consequences that distinguish ICS risks from those in conventional IT systems. The case study results suggest that CVSS v4.0 is more suitable than CVSS v3.1 for ICS vulnerability ranking, because it allows the safety and operational context of the system to be explicitly incorporated into severity assessment. In this case study, it directly changed the prioritization of remediation efforts, escalating a denial-of-service vulnerability from High to Critical once its downstream safety consequences were incorporated. The ability to incorporate system security requirements and subsequent system impact into the scoring allows CVSS v4.0 to better reflect the risk profile of ICS vulnerabilities.

The DFD update stage is more important than it might initially appear. In this case study, design changes between the initial threat model and the testing phase affected approximately one fifth of the identified threats, with 19 threats suppressed as irrelevant and 3 new threats added. Skipping this stage could have resulted in a testing effort misaligned with the actual system, directing resources toward threats that no longer existed while missing threats introduced by implementation decisions.

The case study also highlights the consequences of excluding security requirements testing and threat mitigation testing from the verification and validation process. In this project, due to resource and time constraints, penetration testing was prioritized over security requirements testing and threat mitigation testing. Vulnerability testing was conducted partially, limited to known vulnerabilities on selected system components, rather than the more extensive assessment that a fully resourced project would require. This may explain why several well-specified security requirements that had been defined during the design phase had not been correctly implemented or enforced by the time penetration testing was conducted. Performing security requirements testing and threat mitigation testing before penetration testing, when resources allow, increases the likelihood that defined requirements are verified before the adversarial test is conducted, enabling penetration testers to focus on more advanced attack scenarios rather than finding vulnerabilities that should have been addressed earlier in the process.

6.5 Summary

This chapter presented the BRIDGE methodology, which addresses research gap G7 by providing a structured approach for translating design-phase threat modeling outputs into targeted penetration testing activities. BRIDGE defines nine stages spanning the design, implementation, and testing phases of the ICS lifecycle, with a bidirectional feedback loop that allows penetration testing to serve as a partial validation mechanism for the design-phase threat model. The microgrid case study demonstrated that 9 of 12 reported vulnerabilities had mitigations already specified in the design-phase security requirements, indicating that, in this case, the gap between design and testing was related to implementation and enforcement rather than to a lack of foresight during design. The analysis also showed that CVSS v4.0 provided more contextually relevant severity rankings than CVSS v3.1 for the vulnerabilities in this case study.

7 Integrating Threat Modeling with Cyber Resiliency Assessment for ICS

This chapter addresses RQ4, which asks how threat modeling can be integrated with cyber resiliency assessment frameworks in ICS and what practical challenges arise when operationalizing this integration. It is based on Publication VI [30] and extends its analysis with a detailed examination of the challenges encountered during operationalization. The chapter first presents the motivation and conceptual background, then describes the framework selection and the CRUC methodology, and finally reports the case study and its key findings.

7.1 Motivation

The investigation presented in this chapter originated from a project task to evaluate whether a network segmentation device adds measurable resilience value to a CPS testbed, described in Section 7.5. The author was required to identify a cyber resiliency assessment framework that provides a structured methodology and a quantitative scoring mechanism for comparing alternative system designs. During the search for a suitable framework, a key criterion was whether the framework already incorporated threat modeling as part of its assessment process, since connecting cyber resiliency assessment to threat characterization was identified as a natural extension of the research topic of this thesis. This criterion ultimately led to the identification of CRUC as a particularly relevant framework.

7.2 Conceptual Background

Resilience has been defined in various ways across different regulatory and technical bodies, reflecting the absence of a universally agreed definition [38]. Common to many definitions is the ability of a system to anticipate, absorb, adapt to, and recover from disruptions [38]. These definitions, however, address disruptions broadly, encompassing natural disasters, equipment failures, and other high-impact low-frequency events, without specifically accounting for the intentional and adaptive nature of cyber adversaries. Cyber resiliency extends this concept to focus specifically on the ability of a system to anticipate, withstand, recover from, and adapt to adverse conditions caused by cyber threats and attacks [39, 9].

The design of cyber resiliency assessment frameworks is shaped by how cyber resiliency metrics relate to other metric types, as illustrated in Figure 10. Security metrics assess how well a system provides key security functions and meets objectives such as confidentiality, integrity, and availability. Resilience metrics assess the level of system performance before, during, and after a disruption. Risk metrics assess the likelihood of an adversary attack scenario, the likelihood that a given set of adversary Tactics, Techniques, and Procedures (TTP) succeeds, and the severity of its consequences. Cyber resiliency metrics do not form a separate category. Instead, they repurpose these three metric types by applying a mission-centric and adversary-aware framing. When repurposing resilience metrics, they do not assume that a disruption is detected and instead consider whether an observed anomaly could be the result of an adversary. When repurposing risk metrics, they assume the presence of zero-day vulnerabilities, which makes likelihood assessment of limited value and shifts the focus to the impact of an attack [10].

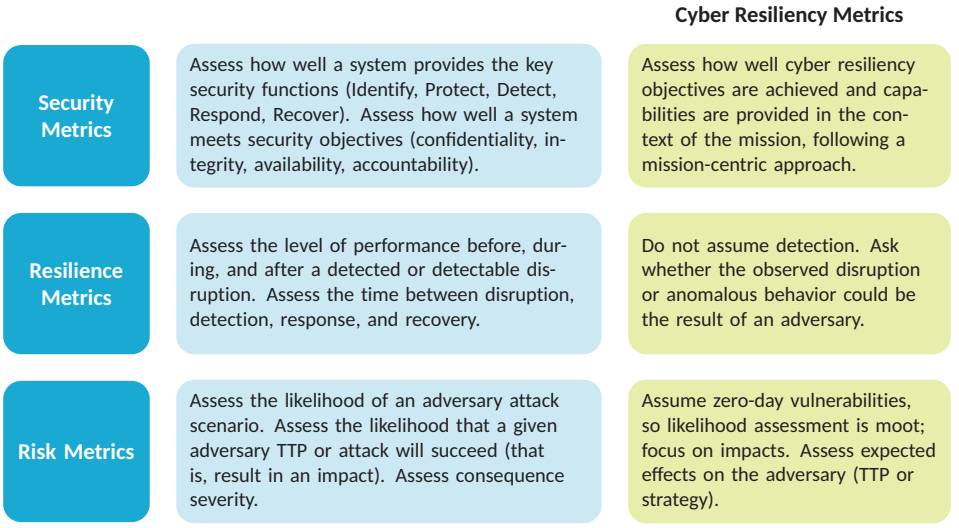


Figure 10: Cyber resiliency metrics repurpose security, resilience, and risk metrics by applying a mission-centric and adversary-aware framing. Adapted from [10].

7.3 Framework Selection

A review of existing resilience metrics and assessment approaches was conducted to identify the most appropriate framework for this investigation. The Grid Modernization Laboratory Consortium (GMLC) framework [7] provides resilience metrics organized into two categories: metrics based on Multi-Criteria Decision Analysis (MCDA), which assess resilience properties such as robustness, adaptiveness, and recoverability through weighted scoring, and performance-based metrics, which quantify the impact of a specified disruption using historical data or computational models. However, this framework provides metrics for quantifying resilience without prescribing an assessment methodology or specifying how threat characterization should be produced, and its focus is on resilience rather than cyber resiliency specifically.

This review revealed the importance of the distinction between resilience and cyber resiliency introduced in Section 7.2, and directed the search toward frameworks that explicitly address cyber threats and incorporate threat characterization as part of the assessment process. The NIST cyber resiliency analysis process described in SP 800-160 Vol. 2 provides a comprehensive but high-level process for understanding context and threat [39], without a structured scoring mechanism for comparing alternative solutions. The MITRE Cyber Resiliency Use Case methodology [9] was identified as a particularly relevant methodology for three reasons. First, it explicitly includes a threat model component as one of its five components. Second, it is aligned with the NIST CREF [39] and provides a repeatable process for comparing alternative solutions. Third, it includes a quantitative scoring mechanism through the Situated Scoring Methodology for Cyber Resiliency (SSM-CR), which uses the CREF elements, including objectives, sub-objectives, and activities, to assess a system's cyber resiliency and compare alternative solutions [10].

7.4 The CRUC Methodology

The CRUC methodology identifies five components, shown in Figure 11. The programmatic and system use concept defines the system type and specifies primary, secondary, and

supporting missions and their criticality. The architecture component describes the high-level system design and identifies external interfaces and dependencies. The threat model characterizes adversaries whose actions could degrade mission execution. The alternative solutions component identifies approaches for improving resilience. The scores, metrics, and measures of effectiveness component defines the scoring framework for assessing the baseline system and comparing alternatives [9].



Figure 11: The five components of the Cyber Resiliency Use Case methodology, adapted from [9].

The threat model component is the point at which CRUC intersects directly with the work carried out in this thesis. CRUC requires the threat model to describe both the characteristics of adversaries, namely their goals and intended effects, and their behaviors, described as TTP [9]. The nine-stage threat modeling methodology developed in DSR Cycle 1 and presented in Chapter 5 [24] produces an attack taxonomy, threat actor identification, a prioritized threat list, and a consequence and loss analysis; however, it does not by default produce TTP-level behavioral descriptions. The STRIDE-to-ATT&CK mapping introduced in BRIDGE and presented in Chapter 6 provides this missing link, translating STRIDE-categorized threats into corresponding ATT&CK techniques. Combining the outputs of the nine-stage methodology with the BRIDGE mapping would therefore more fully satisfy the requirements of CRUC's threat model component. Adapting this combined output to the mission-centric framing of CRUC would require one additional step, namely incorporating mission impact assessment into the consequence and loss mapping stage of the nine-stage methodology [24] to map identified threats to their impact on the defined mission objectives and sub-objectives.

7.5 Case Study

This section describes the testbed on which the CRUC methodology was operationalized, the sequence followed during operationalization, and the methodological observations that emerged from the process.

7.5.1 Context and Testbed

The case study was conducted on a testbed emulating the Information and Communication Technology (ICT) infrastructure of a fictional manufacturing company, built as part of a project evaluating whether a rapid Layer-1 network segmentation device adds measurable resilience value to a CPS. The testbed implements a four-layer network hierarchy comprising a field layer (FIELD), an operational technology layer (OT), an enterprise layer (ENT), and a wide area network layer (WAN). The WAN layer simulates public-origin traffic through two dedicated attacker hosts, providing a controlled environment for executing adversary scenarios without affecting real infrastructure. The ENT layer hosts the internet-facing web server and database. The OT layer contains a Supervisory Control and Data Acquisition (SCADA) system and an energy management system. The FIELD layer contains Programmable Logic Controllers (PLC), a battery energy storage system, a generator set, and a meter that emulate the manufacturing process and its supporting power system [30].

7.5.2 Operationalization

The CRUC methodology was operationalized following the sequence illustrated in Figure 12. As Publication VI [30] is a conference paper with limited length, this section provides additional context on the operationalization process beyond what is reported in the paper itself, while remaining consistent with its content. The first step established the experimental setup and system architecture described in Section 7.5.1. The second step selected the mission goal, objectives, and sub-objectives using the CREF Navigator [33, 30], and defined the activities (referred to as actions in Publication VI [30]). SSM-CR [9, 10] was used to assign weights at the objective, sub-objective, and activity levels and aggregate them hierarchically into the overall resiliency index. Publication VI [30] reports the sub-objective weights used in this case study as an example of how weight scores were assigned across the CRUC structure. Table 6 summarizes these sub-objectives alongside their related objectives, official CREF names, assigned weights, and associated activities/actions.

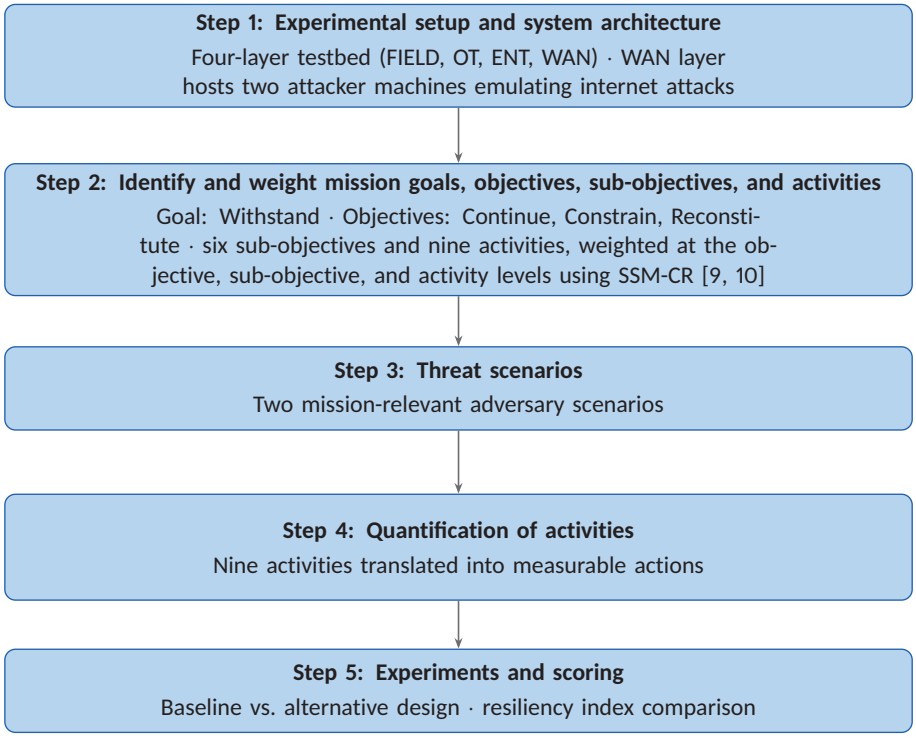


Figure 12: The operationalization sequence followed in this study [30].

In the third step, threat modeling was used to derive mission-relevant adversary scenarios, anchoring the testing and metric design. Two scenarios were defined. In the first, an external attacker carries out a denial-of-service attack against the web application exposed to the internet. In the second, an external attacker pivots from the internet to the OT layer to disrupt the monitoring capabilities of the OT client. In the fourth step, the activities/actions identified in the second step, summarized in Table 6, were made measurable by defining specific quantification formulas for each of them.

In the fifth step, both scenarios were executed against two system configurations: a baseline design and an alternative design with the Layer-1 network segmentation device.

Table 6: Mapping of mission objectives and sub-objectives (SO) used in the CRUC operationalization, with sub-objective weights, labels, and actions as reported in [30], alongside the corresponding official CREF terminology for sub-objectives.

SO#	Related Objective	SO Label [30]	Official CREF SO name [33]	SO Weight	Activities/Actions [30]
SO1	Continue	Interruption	Minimize interruptions in service delivery	5	• Interruption of Manufacturing • Interruption of Power Supply
SO2	Continue	Degradation	Minimize degradation of service delivery	3	• Degraded Communications • Degraded Internet Connectivity
SO3	Continue	Correctness	Ensure that ongoing functioning is correct	3	• Manufacturing Integrity Compromised • Power System Integrity Compromised
SO4	Constrain	Isolate	Isolate resources to limit future or further damage	5	• Isolate Field Layer Devices
SO5	Constrain	Change resource use	Change or remove resources and how they are used to limit further damage	3	• Curtail Loads to Preserve Energy
SO6	Reconstitute	Restore functionality	Restore functionality	5	• Functional Availability

In the alternative configuration, the segmentation device replaces baseline incident response procedures with rapid Layer-1 segmentation and restoration, used to isolate the ENT, OT, and FIELD layers and the relevant assets, specifically the web server in the first scenario and the energy management system and SCADA system in the second. The resiliency index was computed for each configuration under each scenario, allowing the resilience contribution of the segmentation device to be measured directly rather than assumed.

In preparation for the case study, an attempt was made to apply the nine-stage threat modeling methodology to the testbed system, following the sequence of CRUC components shown in Figure 11, in order to produce the threat model component as input to the cyber resiliency assessment. This included the definition of information asset categories and the mapping of the data flows between all system components in two iterative versions. It became clear during this process that applying the nine-stage methodology [24] was not proportionate to what the case study required. A full threat modeling exercise is designed to produce a comprehensive threat characterization across an entire system, as it did for the microgrid in Chapter 5, where it produced 110 structured threats. However, this cyber resiliency assessment was conducted on a testbed and required a limited number of mission-relevant adversary scenarios to compare two design alternatives. Producing the full threat catalog would have required significant time and human resources. It would also have generated extensive detail that the resiliency scoring would not use in this project context. The controlled nature of the testbed reinforced this mismatch, since it had no organizational stakeholders, regulatory requirements, or history of design decisions that a full threat modeling exercise is intended to account for, and the scope of the resiliency evaluation had already been fixed before the threat modeling exercise began.

An alternative sequence was therefore adopted. Rather than completing the full threat modeling exercise first and then deriving mission-relevant scenarios from it, the mission objectives and their weights were defined first using the CREF structure. The threat scenarios were then derived to assess the defined mission goals and objectives. This sequence differs from structured threat modeling practice. Structured threat modeling begins from a representation of the system and elicits threats systematically across all components before prioritizing them by consequence. The CRUC-oriented sequence instead begins from mission priorities and derives threats that assess those priorities

directly, which produces a narrower, mission-centric set of threats.

7.5.3 The Appropriate Level of Effort for Threat Modeling

A methodological observation that emerged from the operationalization concerned the appropriate level of effort for producing the threat characterization that CRUC requires. In this case study, threat modeling was used specifically to derive mission-relevant scenarios rather than to produce a comprehensive threat catalog. This was proportionate to the scope of the evaluation, since the cyber resiliency assessment required a limited number of adversary scenarios to compare two design alternatives.

This stands in direct contrast to the situation that would apply if the same evaluation were conducted on the microgrid system studied in Chapters 5 and 6. For that system, a full threat modeling exercise had already been carried out as part of the design phase. The DFD, the threat list, the consequence analysis, and the prioritization results are all documented and accessible. Integrating those artifacts into the CRUC threat model component would require one additional step, namely extending the existing consequence and loss analysis to include mission impact assessment by mapping the identified threats to their impact on the defined mission objectives and sub-objectives. When design-phase threat modeling artifacts do not exist, as in this case study, or when the goal is specifically to test or demonstrate the cyber resiliency assessment itself rather than to produce a comprehensive security baseline, producing a full structured threat characterization from scratch solely to feed the assessment may not be proportionate. The boundary condition identified by this study is therefore not that threat modeling is incompatible with cyber resiliency assessment, but that the appropriate form of threat modeling input depends on whether design-phase threat modeling work has already been performed for the system under evaluation, and on whether the objective is to produce a comprehensive security baseline or specifically to test the cyber resiliency assessment process itself.

7.6 Key Findings on Boundary Conditions

The CRUC study produces three findings relevant to the integration of threat modeling with cyber resiliency assessment frameworks in ICS [30]. The first finding concerns the identification of an appropriate framework. Among the frameworks considered, CRUC was identified as a particularly relevant framework because it explicitly includes a threat model component, provides a mission-centric scoring approach, and is aligned with the NIST CREF [39].

The second finding concerns the specification of the threat model component within CRUC. The documentation available for CRUC is extensive and distributed across multiple reference documents, and the example threat model provided in the CRUC documentation consists of high-level adversary scenario descriptions without a structured asset identification, a DFD, or a repeatable elicitation procedure. For a practitioner familiar with structured threat modeling approaches, this creates ambiguity about what the component actually requires [9]. In this study, this ambiguity was addressed by establishing mission objectives and weights before deriving threat scenarios, as described in Section 7.5.

The third finding identifies the practical boundary condition and introduces a distinction between two integration approaches. A threat modeling-driven approach, in which existing design-phase threat modeling artifacts feed directly into the CRUC threat model component, is practical and requires limited additional investment when those artifacts already exist, since they can also support security testing, requirements verification, and cyber resiliency assessment. A threat modeling-based approach, in which lightweight adversary scenarios are derived from mission objectives without a prior full threat modeling

exercise, is more appropriate when the evaluation scope is already operationally defined and no design-phase threat modeling artifacts exist. This is the approach adopted in this study. The practical implication is that organizations conducting cyber resiliency assessments should first determine whether design-phase threat modeling has already been performed for the system under evaluation, and whether the objective is to produce a comprehensive security baseline or specifically to test the assessment process itself, as this determines which integration approach is proportionate.

7.7 Summary

This chapter addressed the fourth research question by investigating how threat modeling can be integrated with cyber resiliency assessment frameworks in ICS. The conceptual background established the distinction between resilience and cyber resiliency, and explained how cyber resiliency metrics repurpose existing metric types through a mission-centric and adversary-aware framing. Among the frameworks considered, CRUC was identified as a particularly relevant framework because it explicitly includes a threat model component, is aligned with the NIST CREF [39], and provides a quantitative scoring mechanism. The operationalization of CRUC on a CPS testbed revealed that its threat model component is insufficiently specified, and the case study addressed this by establishing mission objectives and weights before deriving threat scenarios. The study concluded that a threat modeling-driven approach is practical when design-phase threat modeling artifacts already exist, while a threat modeling-based approach using lightweight scenario derivation is more appropriate when the evaluation scope is already operationally defined or when the objective is to test the cyber resiliency assessment process itself [30].

8 Discussion

This chapter synthesizes the findings of the thesis across all four research questions and discusses their theoretical and practical implications. Section 8.1 summarizes how each research question was answered, Sections 8.2 and 8.3 present the theoretical contributions and practical implications, and Section 8.4 reflects on the limitations of the work. Together, these sections position the contributions of the thesis in relation to the research gaps identified in Chapter 4.

8.1 Synthesis of Findings

The first research question asked what the current state and limitations of threat modeling approaches in ICS are. The SLR [25] conducted in Chapter 4 analyzed 36 papers published between 2012 and 2023 and found that STRIDE combined with DFDs dominates the reviewed literature. However, its application is inconsistent across the reviewed studies, which vary in format, level of detail, and the choice between STRIDE-per-element and STRIDE-per-interaction without documenting the rationale behind these choices, making existing work difficult to replicate. Stage coverage is heavily concentrated in the early stages of the process. All selected papers address system mapping and threat enumeration, while threat prioritization, security requirement derivation, and validation receive substantially less coverage. The review identified seven research gaps: limited procedural coverage of the threat modeling process, the absence of a common standard for applying and documenting STRIDE in ICS, undocumented technical and organizational challenges of applying threat modeling in practice, the lack of published industrial case studies, insufficient attention to communication across multidisciplinary threat modeling teams, the lack of rigorous validation for both threat modeling methods and threat modeling outcomes, and the exclusive application of threat modeling in the design phase without extension to later stages of the SDLC. These gaps directly motivated the contributions of this thesis.

The second research question asked how threat modeling can be systematically structured and applied to CPS, considering both cyber and physical aspects. The nine-stage methodology [24] developed and presented in Chapter 5 addresses this directly. The methodology provides a documented, step-by-step procedure covering all stages from attack taxonomy development through asset identification, DFD construction, security context definition, trust boundary determination, STRIDE threat elicitation, consequence and loss mapping, threat prioritization, and security requirement derivation. The DFD notation was enhanced to represent physical processes outside the trust boundaries, making it possible for power systems domain experts and cybersecurity practitioners to jointly trace the consequences of cyber threats into the physical domain. The methodology was applied in full to a real-world microgrid during its design phase, producing 110 structured threats across all six STRIDE categories, 13 threat consequence definitions mapped to 10 prioritized loss types, and a set of security requirements derived from IEC 62443.

The third research question asked how design-phase threat modeling outcomes can guide the penetration test planning and vulnerability assessment of newly deployed ICS. The BRIDGE methodology [26] presented in Chapter 6 addresses this by providing a nine-stage, threat modeling-driven penetration testing process that uses the design-phase threat modeling artifacts as its primary input. A key contribution of BRIDGE is a mapping of STRIDE threat categories to MITRE ATT&CK for ICS techniques, which bridges the abstraction gap between high-level design-phase threat analysis and the intermediate-level attack representations needed for test planning. The methodology was applied to the same microgrid system after its deployment, through a collaborative process involving

five stakeholders over eight months. The most significant finding was that mitigations for 9 of the 12 reported vulnerabilities had already been specified in the design-phase security requirements. In these nine cases, the vulnerabilities existed because the requirements had not been correctly implemented or enforced, rather than because they had not been anticipated during design. A similar link between penetration testing results and design-phase controls was illustrated by Casola et al. on a demonstration cloud application, where two vulnerabilities found through semi-automated testing were traced to input validation and output filtering controls that had been selected during the risk analysis and had not been correctly implemented [11]. The analysis also showed that CVSS v4.0 produced more contextually appropriate severity ratings than CVSS v3.1 for the 12 vulnerabilities reported in the case study, as it explicitly incorporates safety implications and downstream operational consequences into the scoring.

The fourth research question asked how threat modeling can be integrated with cyber resiliency assessment frameworks in ICS, and what practical challenges arise when operationalizing this integration. Chapter 7 provides the answer through an application of the MITRE CRUC methodology [9] to a CPS testbed. CRUC was identified as a particularly relevant framework for this integration because it explicitly includes a threat model component and is aligned with the NIST CREF [39]. The study found that CRUC does not provide sufficient guidance for producing its threat model component in a systematic way. The CRUC documentation is distributed across multiple reference documents. Its example threat model consists of high-level adversary scenario descriptions without a structured asset identification, a DFD, or a repeatable elicitation procedure [9]. The study proposes a distinction between two integration approaches, which differ in whether design-phase threat modeling artifacts already exist. The first is a threat modeling-driven approach, in which existing artifacts feed directly into CRUC with limited additional investment. The second is a threat modeling-based approach, in which lightweight adversary scenarios are derived from mission objectives when no prior threat modeling exists and the evaluation scope is already operationally defined [30].

8.2 Theoretical Contributions

The first contribution of the thesis is a structured analysis of the state of threat modeling in ICS and CPS, grounded in a systematic review of 36 papers. The review introduces a taxonomy of ICS threat modeling methods covering method type, approach, focus, validation method, and application area, and a set of six threat modeling stages used to evaluate lifecycle coverage across the selected papers. The stage coverage analysis makes visible, in a structured and comparable way, which stages of the threat modeling lifecycle the existing approaches address and which receive limited coverage. The seven research gaps identified by the review are not a simple enumeration of missing features; they may help explain why the field has not yet produced a practitioner-ready approach despite more than a decade of published research [25].

The second contribution is the nine-stage threat modeling methodology itself. Although it builds on well-established techniques (STRIDE, DFDs, and IEC 62443), its theoretical significance comes from how it combines them into a documented, repeatable, and empirically demonstrated procedure. This procedure covers the full threat modeling process, from system representation to security requirement derivation. In contrast to the limited number of real-world applications identified in the SLR [25], the methodology was applied end-to-end to a real-world system during its design phase and documented in detail to support independent replication. The DFD enhancements contribute to addressing Gap G5, which concerns insufficient attention to communication across multi-

disciplinary threat modeling teams. The representation of physical processes outside the trust boundaries and the introduction of a color-coding scheme make it easier for power systems experts to follow the DFD and to trace how cyber threats lead to physical consequences [24].

The third contribution is the BRIDGE methodology. Its theoretical significance rests on two elements. The first is a conceptual distinction introduced in this thesis between threat modeling-driven penetration testing, in which a comprehensive design-phase threat model serves as the primary input to the testing process, and threat modeling-based penetration testing, in which a simplified threat model is created during the testing phase. To the best of the author's knowledge, the use of design-phase threat modeling artifacts as the primary driver of penetration test planning in ICS had not been proposed or demonstrated in the literature prior to this work. BRIDGE is a threat modeling-driven approach, and this distinction has practical consequences: the design-phase threat model guides the planning of the penetration tests, and the penetration testing results are in turn used to verify whether the design-phase security requirements were correctly implemented and to validate whether the prioritized threats were effectively mitigated by these requirements. The second element is the STRIDE-to-ATT&CK mapping, which bridges two levels of abstraction in the ICS security lifecycle and provides the traceability mechanism that makes this bidirectional feedback loop possible [26].

The fourth contribution is the empirical analysis of boundary conditions for threat modeling integration with cyber resiliency assessment frameworks. The ICS threat modeling studies selected in Publication I do not discuss the technical and organizational challenges of applying threat modeling or how to overcome them [25], and limited attention has been given to when threat modeling should and should not be applied, or to how its applicability varies across evaluation contexts. This study proposes a distinction between a threat modeling-driven approach, in which existing design-phase artifacts feed directly into the resilience assessment framework with limited additional investment, and a threat modeling-based approach, in which lightweight adversary scenarios are derived from mission objectives when no prior threat modeling exists. The finding that the choice between these two approaches depends on whether design-phase threat modeling work has already been done may help practitioners decide how to allocate resources [30].

8.3 Practical Implications

Many existing threat modeling approaches are demonstrated on theoretical examples or controlled testbed environments [25]. In contrast, the nine-stage threat modeling methodology [24] was applied in full to an AC microgrid during its design phase as part of a real development project, with cybersecurity experts, a software architect, and a power systems expert acting as the system owner, who were all engaged throughout the process. As a result, the procedural steps, documentation requirements, and consequence mapping techniques of the methodology were exercised under the practical constraints, time pressures, and stakeholder coordination challenges of an industrial project rather than under idealized conditions. The DFD enhancements, particularly the inclusion of physical processes outside the trust boundaries, were developed to support this collaboration, allowing domain engineers without a cybersecurity background to engage directly with the threat analysis.

Beyond this direct application, the methodology has been used as a teaching and learning exercise in a graduate course on CPS security at Tallinn University of Technology across four consecutive years. In 2022 and 2023, under the course name *ITC8112 Special Course in Cyber Security 2*, two cohorts of fewer than ten students each applied the

methodology to a simplified network diagram of the microgrid system, using the microgrid attack taxonomy as a guide to threat elicitation. This course was taught in the third semester of the cybersecurity MSc programme.

Following the introduction of the dedicated course *ITC8090 Fundamentals of Cyber-Physical Systems Security*, a pass-fail three-credit course that serves as the only introduction to OT security in the programme, two further cohorts of approximately 20 students applied the methodology in 2024 and 2025 to the manufacturing company testbed developed for the CRUC resiliency assessment project, using the STRIDE-to-ATT&CK for ICS categorization trees developed as part of the BRIDGE methodology to guide threat elicitation, and constructing multi-step attack scenarios using the attack precondition stages introduced in this thesis. The exercise is the concluding group activity of the course and is taken by first-semester students who are still building foundational knowledge in cybersecurity. The repeated application across four cohorts, two system contexts, and two versions of the exercise suggests that the methodology and its associated artifacts are teachable and practically accessible without requiring prior specialization in ICS security.

The methodology has also been adopted beyond its original ICS context. Independent researchers applied the methodology to cross-border payment systems, adapting the first seven stages to model cyber threats across four payment models, and explicitly noted that no standard procedure for applying STRIDE to this domain exists [5]. This suggests that the undocumented-procedure problem identified in the SLR may not be specific to ICS, and that the documented, step-by-step structure of the methodology can be useful beyond the ICS domain.

BRIDGE addresses the practical problem of disconnection between design and deployment. It was evaluated through an actual penetration test conducted on the same deployed microgrid system rather than a simulated exercise. The finding that 9 of the 12 vulnerabilities reported by the penetration tester had mitigations already specified in the design-phase requirements carries a direct message for practitioners: the gap between design and secure deployment is not necessarily a gap in foresight; it may also be a gap in implementation and enforcement. BRIDGE provides a structure to detect this gap and to support its closure. The STRIDE-to-ATT&CK mapping produced as part of BRIDGE can be reused: teams that have conducted a STRIDE-based threat modeling exercise on an ICS can use the mapping as a starting point to translate their threat list into a structured test plan, rather than developing the correspondence from the beginning [26].

The CVSS v4.0 finding has a practical implication for ICS teams scoring vulnerabilities during security assessments. Publication V recommends adopting CVSS v4.0 rather than CVSS v3.1 for vulnerability assessments in ICS [26]. CVSS v4.0 explicitly models subsequent system impact and safety-related consequences. In the microgrid case study, a denial-of-service vulnerability was rated 7.4 (High) under CVSS v3.1. Under CVSS v4.0, it was escalated to 9.2 (Critical) once the safety and operational consequences identified during the design phase were incorporated [26]. This shows that the choice of scoring system can change severity rankings in practice.

The boundary condition finding from the CRUC study, derived from a real project context with defined evaluation requirements, has a practical implication for resource allocation decisions. Organizations that have already conducted design-phase threat modeling on a system may be able to reuse those artifacts for cyber resiliency assessment, penetration testing, and requirements verification. Organizations that have not done so, and that are considering whether to conduct threat modeling specifically for cyber resiliency assessment purposes, should evaluate whether their evaluation scope is already operationally defined and whether a full nine-stage exercise is proportionate to that scope [30].

8.4 Limitations

The primary limitation of the thesis is that the real-world evaluation was limited to a single system. The nine-stage methodology was developed and demonstrated on an AC microgrid, and the BRIDGE methodology was applied to the same system. The SLR identified the scarcity of real-world case studies as a key gap in ICS threat modeling [25]. This thesis addresses this gap by applying both methodologies to a real-world system with the involvement of the system owner and domain experts. This setting provides a stronger empirical foundation than the theoretical examples or testbed environments used in many of the studies reviewed in the SLR [25], as these cannot fully reflect the organizational and project constraints of a real deployment. However, a single energy-sector deployment is not a sufficient basis for claiming general applicability across all ICS domains, and the extent to which the nine-stage methodology and BRIDGE generalize to other domains, such as water treatment, manufacturing, or transportation, remains to be established empirically, which is outlined as a direction for future work in Section 9.2.

The evaluation of both methodologies relies on qualitative case study research, which is consistent with the DSR approach adopted in this thesis and suited to the time and resource limitations of the real development project in which the case studies were conducted. As a result, neither methodology was compared with an alternative approach on the same system, and the effort required was not measured against a baseline. The quantitative results reported in this thesis, such as the 110 threats elicited for the microgrid system [24], therefore describe what the methodologies produced in this context rather than how they perform relative to other approaches.

The requirement of the nine-stage methodology for a multidisciplinary team is both a strength and a limitation. The consequence mapping and loss prioritization stages depend on domain expertise that cybersecurity specialists alone cannot provide. The methodology therefore requires domain engineers (power systems engineers in the microgrid case), system owners, and cybersecurity practitioners to work together throughout the process, and where this sustained collaboration is not possible, the methodology cannot be applied as described. This requirement follows from the nature of ICS and CPS rather than from the proposed approach, as threat modeling for these systems depends on expertise in both the cyber and physical domains [25]; however, it limits the methodology's accessibility to organizations with the resources to assemble and sustain the required team. In addition, as the consequence mapping and loss prioritization stages rely on the judgment of the domain experts and the system owner, their results may vary with the composition and experience of the team.

The SLR covers publications from 2012 to 2023. The ICS threat modeling literature has continued to develop since then, and some of the gaps identified by the review may have been partially addressed in more recent work. The research contributions of the thesis should be understood in the context of the literature as it stood at the time of the review rather than as definitive claims about the current state of the field. In addition, the SLR search terms focused on threat modeling and did not include penetration testing terms, so studies framed mainly around penetration testing may not have been identified. For BRIDGE, this was complemented by the structured search described in Chapter 6, which combined both sets of terms; however, it did not follow the full systematic review protocol of the SLR. Similarly, the frameworks considered in Chapter 7 were identified through a non-systematic review of cyber resiliency assessment approaches.

The BRIDGE finding that 9 of 12 vulnerabilities had mitigations specified in design-phase requirements rests on a single case study, and the proportion of vulnerabilities covered by design-phase requirements may differ in other systems and projects. The result

should be understood as evidence that the BRIDGE approach can help close the design-to-deployment gap in the specific context studied, not as a general claim about how often design-phase requirements cover deployment vulnerabilities across all ICS environments and project types. Similarly, the comparison of CVSS v3.1 and CVSS v4.0 is based on 12 vulnerabilities from a single system, so the observation that CVSS v4.0 produced more contextually appropriate severity ratings is limited to this case.

The boundary condition identified in the CRUC study was derived from a single testbed rather than a system in real operation. The condition follows logically from the scope of the assessment, which was limited and defined in advance to compare two design alternatives using attack scenarios. However, the effort that a full threat modeling exercise would have required was estimated from its partial application, which was discontinued before the threat elicitation stage, rather than measured. The boundary condition therefore remains to be confirmed in resilience assessments of systems in operation.

9 Conclusion and Future Work

This thesis set out to address the absence of a structured, repeatable, and empirically evaluated approach to threat modeling in ICS environments. The research comprised two design science research cycles and an exploratory case study on cyber resiliency assessment. The first cycle began with a systematic literature review, which identified the research gaps addressed by this thesis. Through this work, the thesis has shown how threat modeling can be systematically applied across the ICS security lifecycle, from system design through verification and validation testing to resilience assessment, and has identified conditions that affect the practical value of this approach.

9.1 Conclusions

The four contributions of this thesis collectively address the seven research gaps identified in Chapter 4, which together reflect a problem that has persisted in the ICS security field despite more than a decade of research: the absence of a structured and documented approach to threat modeling spanning system design, security testing, and resilience assessment. The systematic literature review confirmed that existing approaches are inconsistent, underdocumented, and rarely validated in real-world settings. The nine-stage methodology addressed this by providing an end-to-end threat modeling procedure for ICS, demonstrated on a real-world system during its design phase. The BRIDGE methodology extended this work by providing a traceable connection between design-phase threat modeling and penetration testing, and the CRUC study examined how threat modeling can be integrated with cyber resiliency assessment.

Taken together, the findings of the case studies indicate that design-phase threat modeling, when conducted rigorously and documented fully, can create a reusable foundation that supports both security verification and validation testing and resilience assessment. The findings also suggest boundary conditions for this approach: its value is likely to be highest when it is applied to a real system with the involvement of domain experts, and its extension to resilience assessment is more practical when design-phase artifacts already exist. These conditions can help practitioners decide when and how to invest in structured threat modeling across the ICS security lifecycle.

9.2 Future Work

A first direction for future work is to apply the nine-stage methodology and BRIDGE to ICS domains beyond energy systems, such as manufacturing, water treatment, and transportation. Each new domain will require an adapted attack taxonomy, and documenting these adaptations systematically would strengthen the evidence for the general applicability of the approach. Testbeds developed for ICS security research across different domains [14] offer a safe environment for this work, as threat modeling and penetration testing can be carried out on them without risk to an operational system. A testbed was also used for the exploratory case study in Chapter 7. Testbeds would complement the real-world case study of this thesis by allowing repeated and controlled applications, which would also support the quantitative evaluation described in the next direction.

A second direction concerns the quantitative evaluation of the nine-stage methodology. The current evaluation is empirical, demonstrated through a real-world case study; however, it does not include formal metrics for evaluating the completeness, correctness, or efficiency of the methodology's outputs. A user study following a similar approach to the validation conducted for LINDDUN [56], measuring how completely and correctly the methodology identifies threats across different ICS and practitioner backgrounds, would

strengthen the evidence base for the methodology and provide a more rigorous foundation for comparing it with alternative approaches.

A third direction concerns tool support. The procedures of both methodologies are documented in detail to support reproducibility: Publication III provides a detailed worked example of the nine-stage methodology [24], and Publication V shares the STRIDE-based categorization of MITRE ATT&CK for ICS techniques together with its rationale [26]. However, the nine-stage methodology as currently described is a manual, expert-driven process. Partial automation, particularly for the DFD construction stage, the STRIDE elicitation stage, and the STRIDE-to-ATT&CK mapping, would reduce the time and resource requirements of the approach and make it more accessible to organizations with limited cybersecurity resources. Large language models represent a promising avenue for supporting threat elicitation and the STRIDE-to-ATT&CK mapping in particular, as they can generate candidate threat lists that domain experts can then review, refine, and validate. Research into how much of the methodology can be supported by tooling or AI-assisted generation without sacrificing the traceability and domain-expert involvement that make it effective would be a valuable contribution.

A fourth direction concerns making the STRIDE-to-ATT&CK mapping developed in BRIDGE more complete and more traceable. In terms of completeness, the current mapping is limited to the ATT&CK for ICS matrix, while ICS environments increasingly incorporate enterprise IT components [49], so a comprehensive mapping would need to extend to the Enterprise matrix and potentially other matrices, depending on the system architecture. In terms of traceability, Publication V identifies the need for a reporting template that links design-phase threat modeling outputs to penetration test planning decisions and exploitation results [26]. Defining this template would make it possible to trace each prioritized threat to the tests planned for it and their results, and to determine the extent to which the prioritized threats were covered by the penetration test.

A fifth direction follows directly from the CRUC boundary condition finding. The study found that the threat model component within CRUC is underspecified and does not correspond to structured threat modeling practice. A natural next step would be to develop an integration procedure between the nine-stage methodology and the CRUC framework. For the threat modeling-driven approach, the procedure would specify which outputs of the nine-stage methodology and the STRIDE-to-ATT&CK mapping of BRIDGE feed into the threat model component of CRUC, which requires both adversary characteristics and adversary behaviors described as TTP. It would also define how the consequence and loss analysis of the nine-stage methodology is extended with a mission impact assessment, which Chapter 7 identified as the additional step required to integrate existing threat modeling artifacts into CRUC. For the threat modeling-based approach, the procedure would describe how lightweight adversary scenarios can be derived from mission objectives in a structured and repeatable way when no design-phase threat modeling artifacts exist. This would address the practical gap encountered in the CRUC study and make the connection between threat modeling and resilience assessment more principled.

A sixth direction concerns the extension of the methodology to address forensic readiness requirements. Publication I argued that reactive controls, such as incident handling, can benefit from the forensic readiness of the system, and that forensic artifact extraction and analysis requirements should therefore be considered early in development [25]. Experts can address these requirements with existing techniques, for example through the Repudiation category of STRIDE; however, current threat modeling approaches do not provide a systematic instrument for deriving forensic readiness requirements [25]. Forensic readiness, as introduced by Tan [52], has two objectives: maximizing the ability of an en-

vironment to collect credible digital evidence, and minimizing the cost of forensics during an incident response. This fits well with design-phase threat modeling, as the same system representation and threat analysis that identify what could go wrong can also inform what data needs to be collected, preserved, and protected to support a future investigation. Extending the nine-stage methodology to produce forensic readiness requirements alongside the security requirements, and applying this extension to the microgrid system, would make forensic readiness a systematic output of design-phase threat modeling.

A seventh direction concerns the extension of this work into system operation. The contributions of this thesis focus on threat modeling during the design and test planning phases of ICS. Once a system becomes operational, however, new threats targeting its assets continue to emerge throughout a lifecycle that often exceeds ten years, as discussed in Chapter 2 [49]. Dynamic cyber threat intelligence could support the operational phase by keeping the identified threats and countermeasures aligned with the evolving threat landscape. Such integration falls outside the scope of the methodologies proposed here; however, it would extend the contributions of this thesis into the operational phase of the system lifecycle.

List of Figures

1 Overview of the research methodology, comprising DSR Cycle 1 (RQ1–RQ2), DSR Cycle 2 (RQ3), and an exploratory case study (RQ4). 19
2 SLR review protocol, including the research questions, search strategy, inclusion criteria, exclusion criteria, and quality criteria [25]..... 23
3 Proposed ICS threat modeling taxonomy used to classify the 36 selected papers [25]..... 24
4 The six threat modeling stages identified through the review, used to evaluate stage coverage across the selected papers [25]. 27
5 The nine stages of the proposed CPS threat modeling methodology [24].... 31
6 Final version of the attack taxonomy for microgrid systems [24]. 33
7 Data Flow Diagram of the AC microgrid system with trust zones defined at Stages 4 and 5 [24]. 35
8 The BRIDGE methodology for threat modeling-driven penetration testing of newly deployed ICS [26]..... 42
9 Comparison of vulnerability severity ratings under CVSS v3.1 and CVSS v4.0 [26]. 45
10 Cyber resiliency metrics repurpose security, resilience, and risk metrics by applying a mission-centric and adversary-aware framing. Adapted from [10]. 49
11 The five components of the Cyber Resiliency Use Case methodology, adapted from [9]. 50
12 The operationalization sequence followed in this study [30]. 51

List of Tables

1 Mapping of research questions to thesis chapters, publications, and contributions. 13
2 Mapping of research questions to research methodology and publications. 18
3 Mapping of identified research gaps to the chapters and publications that address them. 30
4 Information asset categories identified in the AC microgrid system [24]. 34
5 Number of threats elicited per STRIDE category [24]. 37
6 Mapping of mission objectives and sub-objectives (SO) used in the CRUC operationalization, with sub-objective weights, labels, and actions as reported in [30], alongside the corresponding official CREF terminology for sub-objectives. 52

References

[1] IEC 62443-4-1: Security for industrial automation and control systems – part 4-1: Secure product development lifecycle requirements, 2018.

[2] IEC 62443-4-2: Security for industrial automation and control systems – part 4-2: Technical security requirements for iacs components, 2019.

[3] O. Alexander, M. Belisle, and J. Steele. MITRE ATT&CK for industrial control systems: Design and philosophy. Technical report, The MITRE Corporation, Bedford, MA, USA, 2020.

[4] S. Ali, A. Chub, H. Bahsi, and S. M. Khalil. Extending cyber threat modeling with physical dependency analysis in DC microgrids. In *2025 IEEE 66th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)*, pages 1–10, 2025.

[5] A. Amiruddin, O. C. Briliyant, S. Windarta, M. Y. B. Setiadji, and D. F. Priambodo. Enhancing security in cross-border payments: A cyber threat modeling approach. *International Journal on Informatics Visualization*, 9(6):2451–2466, 2025.

[6] A. Amro, V. Gkioulos, and S. Katsikas. Assessing Cyber Risk in Cyber-Physical Systems Using the ATT&CK Framework. *ACM Transactions on Privacy and Security*, 26(2), 2023.

[7] Argonne National Laboratory. Grid modernization: Metrics analysis (GMLC 1.1) – resilience. Technical report, U.S. Department of Energy, 2017.

[8] H. Bahsi, H. O. Dola, S. M. Khalil, and T. Korõtko. A cyber attack taxonomy for microgrid systems. In *2022 17th Annual System of Systems Engineering Conference (SOSE)*, pages 324–331, 2022.

[9] D. J. Bodeau, R. D. Graubart, R. McQuaid, and J. Woodill. Cyber resiliency metrics and scoring in practice: Use case methodology and examples. Technical Report MTR180449, The MITRE Corporation, 2018.

[10] D. J. Bodeau, R. D. Graubart, R. M. McQuaid, and J. Woodill. Cyber resiliency metrics, measures of effectiveness, and scoring. Technical Report MTR180314, The MITRE Corporation, Bedford, MA, 2018. Public Release Case Number 18-2579.

[11] V. Casola, A. De Benedictis, M. Rak, and G. Salzillo. A cloud SecDevOps methodology: From design to testing. In *Quality of Information and Communications Technology (QUATIC 2020)*, volume 1266 of *Communications in Computer and Information Science*, pages 317–331, Cham, 2020. Springer.

[12] Y.-T. Chen and C.-C. Huang. Determining information security threats for an IoT-based energy internet by adopting software engineering and risk management approaches. *Inventions*, 4(3):53, 2019.

[13] A. Cherepanov. WIN32/INDUSTROYER: A new threat for industrial control systems. Technical report, ESET, 2017.

[14] M. Conti, D. Donadel, and F. Turrin. A survey on industrial control system testbeds and datasets for security research. *IEEE Communications Surveys & Tutorials*, 23(4):2248–2294, 2021.

[15] M. Erbas, S. M. Khalil, and L. Tsiopoulos. Systematic literature review of threat modeling and risk assessment in ship cybersecurity. *Ocean Engineering*, 306:118059, 2024.

[16] FIRST. Common vulnerability scoring system version 4.0: Specification document. `https://www.first.org/cvss/v4.0/specification-document`, 2023.

[17] I. Friedberg, K. McLaughlin, P. Smith, D. Laverty, and S. Sezer. STPA-SafeSec: Safety and security analysis for cyber-physical systems. *Journal of Information Security and Applications*, 34:183–196, jun 2017.

[18] D. Granata, A. Iannaccone, R. Nardone, and L. Romano. From systematic threat search to pentesting: Industrial control systems threat models. *Computers & Security*, 170:105000, 2026.

[19] S. Hacks, S. Katsikeas, E. Ling, R. Lagerström, and M. Ekstedt. powerLang: a probabilistic attack simulation language for the power domain. *Energy Informatics*, 3(1):30, 2020.

[20] A. R. Hevner, S. T. March, J. Park, and S. Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, 2004.

[21] M. Howard and S. Lipner. *The Security Development Lifecycle*. Microsoft Press, 2006.

[22] ISA Global Cybersecurity Alliance. Security lifecycles in the ISA/IEC 62443 series: Security of industrial automation and control systems. Technical report, ISA Global Cybersecurity Alliance, October 2020.

[23] Joint Task Force. Security and privacy controls for information systems and organizations. Technical Report NIST SP 800-53 Rev. 5, National Institute of Standards and Technology, 2020.

[24] S. M. Khalil, H. Bahsi, H. O. Dola, T. Korõtko, K. McLaughlin, and V. Kotkas. Threat modeling of cyber-physical systems - a case study of a microgrid system. *Computers & Security*, 124:102950, 2023.

[25] S. M. Khalil, H. Bahsi, and T. Korõtko. Threat modeling of industrial control systems: A systematic literature review. *Computers & Security*, 136:103543, 2024.

[26] S. M. Khalil, H. Bahsi, and T. Korõtko. Bridging industrial control systems design and testing through threat modeling-driven penetration testing – a microgrid case study. *Computers & Security*, 167:104906, 2026.

[27] S. M. Khalil, H. Bahsi, and T. Korõtko. Penetration testing of newly-deployed industrial control systems. In J. Staggs and S. Shenoi, editors, *Critical Infrastructure Protection XIX: 19th IFIP WG 11.10 International Conference, ICCIP 2025, Revised Selected Papers*, volume 763 of *IFIP Advances in Information and Communication Technology*, pages 25–48, Cham, 2026. Springer.

[28] S. M. Khalil, H. Bahsi, and P. Tšikul. Analysis of Windows 10 hibernation file. In *6th Interdisciplinary Cyber Research Conference (ICR 2020)*, page 5, Tallinn, Estonia, 2020.

[29] B. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University, 2007.

[30] T. Korõtko, S. M. Khalil, A. Schuller, V. Kotkas, J. Lambot, and K. Pappel. Implementation of MITRE CRUC for operational resilience assessment in cyber-physical systems. In *2026 IEEE International Conference on Industrial Technology (ICIT2026)*, 2026.

[31] C. C. Lee, T. G. Tan, V. Sharma, and J. Zhou. Quantum computing threat modelling on a generic CPS setup. In *Applied Cryptography and Network Security Workshops: ACNS 2021 Satellite Workshops, AIBlock, AIHWS, AIoTS, CIMSS, Cloud S&P, SCI, SecMT, and SiMLA, Kamakura, Japan, June 21–24, 2021, Proceedings*, pages 171–190. Springer, 2021.

[32] R. M. Lee, M. J. Assante, and T. Conway. Analysis of the cyber attack on the Ukrainian power grid. Technical report, Electricity Information Sharing and Analysis Center (E-ISAC) and SANS ICS, March 2016.

[33] MITRE Corporation. CREF Navigator. Online resource, `https://crefnavigator.mitre.org/navigator/`, 2023.

[34] M. M. Morana and T. UcedaVelez. *Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis*. John Wiley & Sons, 2015.

[35] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45–77, 2007.

[36] B. Ramis Ferrer, S. O. Afolaranmi, and J. L. M. Lastra. 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*, pages 3498–3505, 2017.

[37] E. Rencelj Ling, J. E. Urrea Cabus, I. Butun, R. Lagerström, and J. Olegard. Securing communication and identifying threats in RTUs: A vulnerability analysis. In *Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES 2022)*, Vienna, Austria, 2022. ACM.

[38] W. Rickerson, K. Zitelman, and K. Jones. Valuing resilience for microgrids: Challenges, innovative approaches, and state needs. Technical report, National Association of Regulatory Utility Commissioners and National Association of State Energy Officials, February 2022.

[39] R. Ross, V. Pillitteri, R. Graubart, D. Bodeau, and R. McQuaid. Developing cyber-resilient systems: A systems security engineering approach. Technical Report SP 800-160 Vol. 2 Rev. 1, National Institute of Standards and Technology, 2021.

[40] Q. Rouland, B. Hamid, and J. Jaskolka. Specification, detection, and treatment of STRIDE threats for software components: Modeling, formal methods, and tool support. *Journal of Systems Architecture*, 117:102073, 2021.

[41] P. Runeson and M. Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131–164, 2009.

[42] G. Salzillo, M. Rak, and F. Moretta. Threat Modeling based Penetration Testing: The Open Energy Monitor Case study. *ACM International Conference Proceeding Series*, 2020.

[43] M. J. Sarvejahani. Towards architectural pen test case generation and attack surface analysis to support secure design. In *2025 IEEE 22nd International Conference on Software Architecture Companion (ICSA-C)*, pages 143–148, 2025.

[44] O. Saßnick, T. Rosenstatter, C. Schäfer, and S. Huber. STRIDE-based methodologies for threat modeling of industrial control systems: A review. In *2024 IEEE 7th International Conference on Industrial Cyber-Physical Systems (ICPS)*, pages 1–8, St. Louis, MO, USA, 2024. IEEE.

[45] R. Scandariato, K. Wuyts, and W. Joosen. A descriptive study of Microsoft's threat modeling technique. *Requirements Engineering*, 20(2):163–180, 2015.

[46] J. Searle, G. Rasche, A. Wright, and S. Dinnage. NESCOR guide to penetration testing for electric utilities. *National Electric Sector Cybersecurity Organization Resource*, page 8, 2016.

[47] N. Shevchenko, T. A. Chick, P. O'Riordan, T. P. Scanlon, and C. Woody. Threat Modeling: A Summary of Available Methods. Technical report, Carnegie Mellon University, Software Engineering Institute, July 2018.

[48] A. Shostack. *Threat Modeling: Designing for Security*. John Wiley & Sons, 2014.

[49] K. Stouffer, M. Pease, C. Tang, T. Zimmerman, V. Pillitteri, S. Lightman, A. Hahn, S. Saravia, A. Sherule, and M. Thompson. Guide to operational technology (OT) security. Technical Report NIST SP 800-82 Rev. 3, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2023.

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

[51] E. Süren, F. Heiding, J. Olegård, and R. Lagerström. PatrIoT: Practical and agile threat research for IoT. *International Journal of Information Security*, 22(1):213–233, 2023.

[52] J. Tan. Forensic readiness. *Cambridge, MA:@ Stake*, 1:148, 2001.

[53] K. Tuma, G. Calikli, and R. Scandariato. Threat analysis of software systems: A systematic literature review. *Journal of Systems and Software*, 144(June):275–294, 2018.

[54] F. Valenza, E. Karafili, R. V. Steiner, and E. C. Lupu. A hybrid threat model for smart systems. *IEEE Transactions on Dependable and Secure Computing*, 2022.

[55] A. Vernotte, M. Välja, M. Korman, G. Björkman, M. Ekstedt, and R. Lagerström. Load balancing of renewable energy: a cyber security analysis. *Energy Informatics*, 1(1):1–42, 2018.

[56] K. Wuyts, R. Scandariato, and W. Joosen. Empirical evaluation of a privacy-focused threat modeling methodology. *Journal of Systems and Software*, 96:122–138, 2014.

[57] R. K. Yin. *Case Study Research and Applications: Design and Methods*. Sage, 6 edition, 2018.

[58] I. Zografopoulos, J. Ospina, X. Liu, and C. Konstantinou. Cyber-Physical Energy Systems Security: Threat Modeling, Risk Assessment, Resources, Metrics, and Case Studies. *IEEE Access*, 9:29775–29818, 2021.

Acknowledgements

Sometimes life takes you down paths
you never imagined would be yours.

You may not see what lies ahead,
or believe how far you can go.

Until one day you look back and realize
that every small step has brought you to a place
you once thought was beyond your reach.

You then discover that the journey itself
was more beautiful than anything
you had ever dreamed of.

This thesis is the result of a journey that began long before its first pages were written, and it would not have been possible without the people who were part of it.

I am deeply grateful to my supervisor, Professor ***Hayretdin Bahşi***, for his guidance, patience, and unwavering support throughout both my MSc and PhD at Tallinn University of Technology. A good supervisor does not pave the path; he holds the light so that others can find their own way. This is what Prof. Hayretdin did at every stage of this journey, and I thank him for believing in this work and in me.

I would like to thank Dr. ***Tarmo Korõtko*** for his collaboration and continuous support throughout the research projects that form the basis of this thesis. His expertise in power systems was invaluable to the technical grounding of this work. The research behind the three journal publications included in this thesis was carried out within projects that he led.

The research presented in this thesis was supported by funding acknowledged in the individual publications.

I would also like to thank Professor ***Kieran McLaughlin*** from Queen's University Belfast for reviewing the manuscript of my first journal publication before its submission. His feedback was a valuable learning experience at the start of my doctoral studies.

I am also grateful to Associate Professor ***Rain Ottis*** for his continued support as group manager and for the trust he has always shown through both his words and actions.

I would like to thank Dr. ***Adrian Nicholas Venables*** for his support and encouragement in my role as a lecturer in the Cybersecurity MSc programme at Tallinn University of Technology.

To ***Kristi Ainen***, ***Ülle Ainsoo***, and ***Emily Ridal***: thank you for your kindness and support throughout this journey.

I am grateful to Dr. ***Leonidas Tsiopoulos*** for his internal review of this thesis and his constructive feedback. I also sincerely thank my opponents, Professor ***Jianying Zhou*** and Professor ***Sokratis Katsikas***, for their careful reading of this thesis and their valuable comments.

I also thank ***Toomas Lepik***, ***Mall Kulasalu***, Associate Professor ***Risto Vaarandi***, and

Johanna Kasenurm for their help with the Estonian translations in this thesis.

To everyone whose actions, knowingly or unknowingly, helped shape this journey: thank you. Ten years ago, I could never have imagined this path, but I am deeply grateful for where it has led me.

When I decided to leave Cairo and come to Estonia for my MSc, the journey was made possible by people who trusted my choice and encouraged me from the very beginning: ***Mohammed Ragaei***, ***Ramzy al Masry***, ***John Atef***, ***Aya Mohamed***, ***Yehia Elmaghrabi***, and ***Hani Shamseldeen***. I was unable to acknowledge any of you in my MSc thesis, so please consider this acknowledgement long overdue.

To ***Sara Tosson***, who never stopped calling to ask how I was doing throughout all these years: your caring presence, despite the distance, meant a lot to me.

To my parents, ***Bothayna Maghazy*** and ***Mamdouh Radwan***, who supported not only my dream but also the journey itself: thank you for believing in a path that must have seemed uncertain from where you stood. You made education a priority from my very first day at school, and you taught me to value it as much as you did. You stood behind every step I chose, supported me in every possible way, and cared for Salam during my long working hours in Cairo. You were both teachers, and I never imagined becoming one; yet I discovered how much I enjoy teaching, and I now understand that I learned it from you.

To my sisters, ***Asmaa*** and ***Doaa***: as your older sister, I always felt responsible for our family, and when I left, you took on that responsibility in my place. Thank you for being there across every distance. Asmaa, thank you for caring for our parents when I could not be there; Doaa and my brother-in-law, ***Shehab Mahmoud***, thank you for always stepping in when it mattered most.

To my big family, and to all my friends and colleagues who supported me throughout this journey with kind words and actions: thank you. Each of you has a place in my heart, and I am deeply grateful.

To ***Zeyad Khalaf***, who always believed in me and saw what I could not yet see in myself: you supported us from a distance during the MSc, joined us for the PhD, and survived the late nights and early mornings of work, the food deliveries and takeaways, and the beautiful chaos that a PhD brings into a home. Thank you for your love, your patience, and your support, and for trying to understand me even when we see things differently.

To ***Salam Masry***: you were three years old when we left Cairo together, leaving behind every comfort you knew and the warmth of a big family around you for a much simpler lifestyle, with just the two of us. You sat through my night classes and waited quietly outside the door during my online MSc defence without making a sound. When it was over, you hugged me and said *"they liked your thesis"*. I am hoping for the same hug after this defence. You have been my partner in this journey from the very beginning, and no words can thank you enough. This PhD is yours as much as it is mine.

Abstract

Threat Modeling for Industrial Control Systems: From System Design to Security Assessment

Industrial control systems (ICS) underpin critical infrastructure across the energy, manufacturing, water, and transportation sectors. As these systems become increasingly networked and integrated with information technology components, they are exposed to a growing range of cyber threats with the potential to cause physical damage to equipment, disrupt essential services, and endanger human safety. Despite the recognized value of threat modeling as a design-phase security practice, its application in ICS environments remains immature. This thesis investigates how threat modeling can be systematically applied to ICS, from system design to security assessment.

The thesis follows two cycles of design science research, complemented by an exploratory case study on cyber resiliency assessment. The first cycle began with a systematic literature review of 36 papers, which found that STRIDE combined with data flow diagrams dominates the reviewed literature, although its application is inconsistent and underdocumented. The review identified seven research gaps, including limited procedures for the stages other than system mapping and threat enumeration, and the exclusive use of threat modeling in the design phase, without carrying its outputs forward into security testing. In response, a nine-stage threat modeling methodology for cyber-physical systems was developed and applied end-to-end to a real-world microgrid during its design phase, producing 110 structured threats, an impact-based prioritization of these threats, and a set of security requirements derived from IEC 62443.

In the second cycle, the BRIDGE methodology was developed to carry these outputs forward into penetration testing of newly deployed ICS, using a mapping of STRIDE threat categories to MITRE ATT&CK for ICS techniques. Its application to the same microgrid showed that mitigations for 9 of the 12 reported vulnerabilities had already been specified during the design phase, although they had not been fully implemented or correctly enforced. In addition, the Common Vulnerability Scoring System (CVSS) v4.0 produced more contextually appropriate severity ratings for these vulnerabilities than CVSS v3.1. Finally, the exploratory case study applied the MITRE Cyber Resiliency Use Case (CRUC) methodology to a manufacturing testbed. As the scope of this evaluation did not require a full threat modeling exercise, lightweight adversary scenarios were derived from mission objectives instead. The study therefore proposed a distinction between a threat modeling-driven approach, in which existing design-phase artifacts feed into resilience assessment with limited additional investment, and a threat modeling-based approach, which suits cases where no prior design-phase threat modeling exists.

The case studies presented in this thesis show how threat modeling can be systematically applied across the ICS security lifecycle, from design through penetration testing to resilience assessment. The findings indicate that design-phase threat modeling, when conducted rigorously and documented fully, can create a reusable foundation that supports both security verification and validation testing and resilience assessment. The methodologies developed in this thesis offer practitioners a well-documented way to create this foundation and reuse it across the ICS security lifecycle.

Kokkuvõte
Ohtude modelleerimine tehnojuhtimissüsteemidele: süsteemi kavandamisest turvahindamiseni

Tehnojuhtimissüsteemid (ICS) on kriitilise taristu alustalaks energia-, tootmis-, vee- ja transpordisektoris. Kuna need süsteemid muutuvad üha enam võrgustatuks ning neid integreeritakse infotehnoloogiliste komponentidega, puutuvad nad kokku suureneva hulga küberohtudega, mis võivad põhjustada seadmete füüsilist kahjustumist, häirida olulisi teenuseid ja seada ohtu inimeste elu ja tervise. Vaatamata sellele, et ohtude modelleerimist peetakse juba süsteemide kavandamisetapis oluliseks praktikaks, on selle rakendamine tehnojuhtimissüsteemides endiselt ebaküpsel tasemel. Käesolev doktoritöö uurib, kuidas ohtude modelleerimist saab süstemaatiliselt rakendada tehnojuhtimissüsteemidele alates süsteemi kavandamisest kuni turvahindamiseni.

Doktoritöö põhineb kahel disainiteaduslikul uurimistsüklil, mida täiendab küberkerksuse hindamist käsitlev eksploratiivne juhtumiuuring. Esimene tsükkel algas 36 teadusartiklit hõlmanud süstemaatilise kirjandusülevaatega, millest selgus, et STRIDE koos andmevooskeemidega on analüüsitud kirjanduses domineeriv lähenemisviis, kuigi selle rakendamine on ebaühtlane ja ebapiisavalt dokumenteeritud. Ülevaates tuvastati seitse uurimislünka. Nende hulgas olid puudulikud protseduurid muude etappide jaoks peale süsteemi kaardistamise ja ohtude loendamise ning ohtude modelleerimise kasutamine ainult kavandamisetapis, ilma selle tulemusi turvatestimisse edasi kandmata. Uurimislünkade täitmiseks töötati välja üheksaetapiline ohtude modelleerimise metoodika küberfüüsiliste süsteemide jaoks ning rakendati seda terviklikult reaalse mikrovõrgu kavandamisetapis. Tulemuseks oli 110 struktureeritud ohu tuvastamine, nende ohtude mõjupõhine prioriseerimine ning IEC 62443 standardist tuletatud turvanõuete kogumi koostamine.

Teises tsüklis töötati välja BRIDGE metoodika, et kanda need tulemused edasi äsja kasutusele võetud tehnojuhtimissüsteemide läbistustestimisse. Selleks kasutati STRIDE-i ohukategooriate vastavusse viimist MITRE ATT&CK for ICS tehnikatega. Metoodika rakendamine samal mikrovõrgul näitas, et 12 tuvastatud haavatavusest 9 puhul olid leevendusmeetmed määratletud juba kavandamisetapis, kuigi neid ei olnud täielikult rakendatud ega korrektselt jõustatud. Samuti andis Common Vulnerability Scoring System (CVSS) v4.0 nendele haavatavustele kontekstuaalselt sobivamaid raskusastme hinnanguid kui CVSS v3.1. Lõpuks rakendati eksploratiivses juhtumiuuringus MITRE küberkerksuse kasutusjuhtumi (CRUC) metoodikat tootmissüsteemi testkeskkonnas. Kuna selle hindamise ulatus ei nõudnud täielikku ohtude modelleerimist, tuletati selle asemel lihtsustatud ründestsenaariumid missioonieesmärkidest. Seetõttu pakuti uuringus välja eristus ohtude modelleerimisest lähtuva ja ohtude modelleerimisel põhineva lähenemisviisi vahel. Esimesel juhul toetavad olemasolevad kavandamisetapi artefaktid küberkerksuse hindamist piiratud lisainvesteeringuga. Teine lähenemisviis aga sobib olukordadesse, kus eelnev kavandamisetapi ohtude modelleerimine puudub.

Käesolevas doktoritöös esitatud juhtumiuuringud näitavad, kuidas ohtude modelleerimist saab süstemaatiliselt rakendada tehnojuhtimissüsteemide turvalisuse elutsükli vältel, alates kavandamisest läbi läbistustestimise kuni küberkerksuse hindamiseni. Tulemused osutavad sellele, et põhjalikult läbi viidud ja täielikult dokumenteeritud kavandamisetapi ohtude modelleerimine võib luua korduvkasutatava aluse, mis toetab nii turvalisuse verifitseerimis- ja valideerimistestimist kui ka küberkerksuse hindamist. Käesolevas doktoritöös välja töötatud metoodikad pakuvad praktikutele hästi dokumenteeritud teed sellise aluse loomiseks ja korduvkasutamiseks kogu tehnojuhtimissüsteemide turvalisuse elutsükli vältel.

Appendix 1

I

S. M. Khalil, H. Bahsi, and T. Korõtko. Threat modeling of industrial control systems: A systematic literature review. *Computers & Security*, 136:103543, 2024

Computers & Security 136 (2024) 103543

Contents lists available at ScienceDirect

Computers & Security

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

Threat modeling of industrial control systems: A systematic literature review

Shaymaa Mamdouh Khalil [a,*], Hayretdin Bahsi [a,b], Tarmo Korõtko [c]

[a] School of Information Technologies, Tallinn University of Technology, Estonia
[b] School of Informatics, Computing, and Cyber Systems, Northern Arizona University, United States
[c] FinEst Centre for Smart Cities, Tallinn University of Technology, Estonia

ARTICLE INFO

Keywords:
Threat modeling (TM)
Cyber-physical system (CPS)
Industrial control system (ICS)
Systematic literature review (SLR)
Security-by-design
Risk assessment
Security analysis

ABSTRACT

Threat modeling is the process of identifying and mitigating potential threats to a system. It was originally developed to enhance software security during the design phase but has since been adapted for Industrial Control Systems (ICSs). ICSs are complex and interconnected systems that control critical infrastructure, such as power plants, water treatment facilities, and manufacturing plants. As such, they are major targets for cyberattacks, which may lead to human casualties, severe national security impacts, and financial instability. This systematic literature review explores the existing threat modeling methodologies for ICSs and emphasizes the importance of employing methodical frameworks that cover safety, security, and privacy aspects with clear procedural guidelines. The review reveals that ICSs threat modeling often lacks validation to ensure that the used methodologies are effective in identifying and mitigating threats. This study emphasizes the need to develop and apply better validation metrics in case studies. The main goal of this review is to help cyber security researchers and practitioners in selecting a suitable threat modeling approach that facilitates the creation of ICSs with an acceptable level of security.

1. Introduction

Industrial Control Systems (ICSs) control and monitor industrial processes in critical sectors such as energy (e.g., nuclear plant, smart grid, and oil and gas), chemical, transportation, pharmaceutical, food and beverage, and discrete manufacturing (e.g., automotive, aerospace, and durable goods) (Stouffer et al., 2015a). They are complex systems of systems that use proprietary software, network protocols, specialized standards, and merge Information Technology (IT) with Operation Technology (OT). In the past, ICSs networks used to be isolated, and cyber security was not considered in the early stages of system development. However, a major challenge is currently facing ICSs, due to the fast expansion in the usage of IIoT as a part of Industry 4.0.

ICSs networks are no longer such isolated networks. They are mostly connected to the internet due to functionality and system maintenance requirements. Unlike IT, where the confidentiality of sensitive information is mostly the priority, safety, and operation continuity are fundamentals for OT, which may create conflicts between system requirements and cyber-security. For example, encrypted network protocols might cause latency in the OT network, which could severely impact operational safety. Also, encryption might require replacing old network components with new ones that support such a feature. However, the life cycle of an ICS mostly exceeds 10 years (Stouffer et al., 2015b), and major system component replacement is not financially foreseen. Such paradigm shifts between IT and OT security are major challenges in ICSs security.

To successfully carry out a high-impact industrial control system attack, an attacker must first acquire system access, then exploit a vulnerability. Yet, the likelihood of success of such an attack is reduced if the vulnerability is newly discovered (Allodi and Etalle, 2017). Targeted attacks against ICSs are more likely to succeed, even with new vulnerabilities, as they are often carried out by well-resourced attackers (e.g., nation-state actors, terrorist organizations, and cyber-criminal groups). The 2016 and 2022 Ukrainian power grid attacks, which used the Industroyer malware, showed a sophisticated example of targeted malware that was specially designed to cause power grid damage by abusing ICSs protocols and commands (Cherepanov, 2017; Research, 2022).

* Corresponding author.
E-mail address: Shaymaa.Khalil@Taltech.ee (S.M. Khalil).

https://doi.org/10.1016/j.cose.2023.103543
Received 18 May 2023; Received in revised form 28 September 2023; Accepted 16 October 2023
Available online 20 October 2023
0167-4048/© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

The increasing number of cyberattacks on industrial control systems (ICSs), especially those in critical infrastructure,[1] underscores the need to develop secure-by-design systems that have an acceptable level of security. Threat modeling (TM) is a process of identifying and analyzing potential threats to a system. It is an important step in the secure-by-design process because it helps to identify security requirements early in the development life cycle. This can help to prevent security vulnerabilities from being introduced into the system. However, addressing security in the later stages of development can result in significant engineering expenses or unaddressed critical vulnerabilities. Such potential consequences are more evident in ICSs due to their complex supply chain cycles and less tolerance to major system updates.

In addition to being a task in the development life cycle, threat modeling studies constitute a significant human interaction medium for different development parties at the early stages (e.g., software and hardware developers, system architects, and security experts). The relevant parties can establish common ground in overall security considerations, discuss specific security requirements, and solve potential conflicts between system and security functions in this task.

In the digital world, threat modeling was first introduced for securing software during the design phase, as a part of the secure software development life cycle (SDLC). The use of threat modeling techniques was then extended to cover ICSs as well. Threat modeling is currently included in some ICSs standards. For example, the ISA/IEC 62443 standard[2] discusses threat modeling in the secure product development life cycle requirements (ISA 62443-4-1). However, standards do not provide specific procedures on how to perform threat modeling for ICSs.

Compared to software-based systems, the components of ICSs interact and change the physical space, which makes predicting the physical consequences of cyber attacks more challenging in the threat modeling of such systems. Physical and cyber attack vectors may be interleaved immensely in an attack campaign, making it difficult to dissect the system assumptions and agree on a common ground for the security requirements.

This paper aims to provide an overview of the state of the art of ICSs threat modeling and identify the related research gaps using a Systematic Literature Review (SLR). The research aims to answer the following Research Questions (RQs):

RQ1: What are the proposed threat modeling methods for ICSs in the literature?

RQ1.1: How could we categorize the proposed ICSs threat modeling methods?

RQ1.2: What are the ICSs threat modeling phases that are covered in the literature?

RQ2: What are the identified research gaps in the proposed ICSs threat modeling methods?

This study provides a comprehensive overview of threat modeling for industrial control systems (ICSs). It begins with an extensive background section that introduces readers to the different concepts and terminology related to threat modeling. The study then analyzes a selection of papers on threat modeling for ICSs, based on a proposed taxonomy of ICSs threat modeling and identified threat modeling stages. The study also discusses the state of the art in threat modeling for ICSs, including research gaps. This study is therefore of interest to academic researchers and practitioners who are researching or applying threat modeling to ICSs. To our knowledge, no other study has provided such a comprehensive analysis of threat modeling methods within the scope of ICSs.

The rest of this paper is organized as follows: Section 2 provides background about TM and information that eases the understanding of the topic, especially for readers with limited knowledge about the domain. The background section includes TM definitions and discusses the difference between TM, risk assessment, and attack modeling. Section 2 also introduces STRIDE, the application of TM within the development life cycle, as well as the privacy, safety, and cyber-security focus within TM. The background section concludes with TM tools, gamification, and challenges. Section 3 discusses the related work. Section 4 introduces the research methodology, including the paper selection process, exclusion and inclusion criteria. The categorization criteria, Section 5, are described by an ICSs TM taxonomy and the TM stages. Results of the SLR can be found in Section 6, which includes general information about the selected papers and analyzes them based on the proposed ICSs TM taxonomy and TM stages. The research gaps and future research directions are discussed in Section 7, while Section 8 discusses the threats to the validity of this study. The SLR is concluded by Section 9.

2. Background

This section provides an overview of the concepts of threat modeling, which can help readers get familiar with the topic.

2.1. Threat modeling definitions

The literature shows that the *"definitions of threat modeling are numerous, and used in many different and perhaps also incompatible ways"* (Xiong and Lagerström, 2019). One of the threat modeling definitions promoted by Xiong and Lagerström (2019) is: *"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"*. Nweke and Wolthusen (2020) define TM as *"a systematic approach for characterizing potential threats to a system. It ensures completeness by including the prioritization of threats and mitigation based on probabilities, business impacts, and cost of countermeasures"*.

Threat modeling is a practical process with definitions also coming from the industry and non-academic research. As a part of the Security Development Life cycle (SDL),[3] Microsoft defines threat modeling as *"a practice that allows development teams to consider, document, and (importantly) discuss the security implications of designs in the context of their planned operational environment and in a structured fashion"*. The threat modeling manifesto, created in 2020 by a group of experts from both academic and corporate fields,[4] defines threat modeling as *"analyzing representations of a system to highlight concerns about security and privacy characteristics"*. However, we find that the last definition is more relevant to software TM, as in the context of ICSs, threat modeling could also highlight concerns about the safety characteristics of the system besides its security and privacy. The manifesto also identifies four questions that should be asked during the threat modeling process: (1) What are we working on? (2) What can go wrong? (3) What are we going to do about it? (4) Did we do a good enough job?

2.2. Threat modeling and risk assessment

The term *"risk assessment"* is often used alongside *"threat modeling"* in the literature while referring to threat modeling methodologies. The integration of risk assessment techniques into the threat modeling process helps in better threat prioritization and obtaining more tangible results, as proposed by Maheshwari and Prasanna (2016). However, we noticed that there is no clear distinction between the two terminologies, as they seem to refer to similar concepts in some specific cases, such as asset-centric threat models (Section 5.1.2).

To understand the differences between the two terminologies, we have to distinguish between a *"threat"* and a *"risk"*. Based on NIST guide

[1] https://www.microsoft.com/en-us/security/business/microsoft-digital-defense-report-2022.

[2] https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards.

[3] https://www.microsoft.com/en-us/securityengineering/sdl.

[4] https://www.threatmodelingmanifesto.org/.

to industrial control systems security (Stouffer et al., 2015b), a threat is defined as *"any circumstance or event with the potential to adversely impact agency operations (including mission, functions, image, or reputation), agency assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service"*. The same document defines the risk as *"the level of impact on agency operations (including mission, functions, image, or reputation), agency assets, or individuals resulting from the operation of an information system, given the potential impact of a threat and the likelihood of that threat occurring"*.

Stouffer et al. (2015b) also defined the risk assessment as *"the process of identifying risks to operations, assets, or individuals by determining the probability of occurrence, the resulting impact, and additional security controls that would mitigate this impact"*. Risk assessment is a part of the risk management process, which includes three phases: (i) the conduct of a risk assessment; (ii) the implementation of a risk mitigation strategy; and (iii) the employment of techniques and procedures for the continuous monitoring of the security state of the information system.

Risk assessment provides a clear emphasis on the impact analysis and prioritization of the risk findings, whereas threat models may not necessarily include such constructs. For instance, STRIDE focuses on finding the threats without using an impact terminology and ranking approach, thus, it requires an additional framework (e.g., DREAD) to complement that gap. However, lightweight threat models embed the risk assessment stage into their threat elicitation process to reduce the workload of experts. This can be done by focusing on critical information flows (Tuma et al., 2017), or eliminating threats with lower likelihood and impact (Wuyts et al., 2020). Some research has also proposed incorporating the risk notion directly into threat models (Sion et al., 2018).

The NIST guide for risk assessment (Force, 2012) specifies that risk assessment should be conducted throughout the system development life cycle, starting from the development phase. The guide also states that risk assessment includes identifying threat sources (e.g., adversarial, trusted insider), determining threat events at the level of tactics, techniques, and procedures, and attack graph techniques. We can conclude that threat modeling can also be considered as a form of risk assessment, especially in situations where the risk is calculated. This conclusion is consistent with the NIST definition of threat modeling as *"a form of risk assessment that models aspects of the attack and defense sides of a logical entity, such as a piece of data, an application, a host, a system, or an environment"* (Force, 2017).

2.3. Threat modeling versus attack modeling

The term, *"attack modeling"*, is often used in the literature to refer to the process of identifying security risks from the attacker's perspective (Sequeiros et al., 2020). This does not necessarily distinguish it from the attack-centric threat modeling (Section 5.1.2). However, a nuanced point would be that threat modeling effort always considers the implications of attacks to the defense actions, whereas attack modeling may solely focus on attacker behavior, not necessarily the defense perspective.

Threat models typically include a representation of the target system, such as a data flow diagram, and utilize it as a systematization construct for threat elicitation. Attack models, on the other hand, may not include such a representation. For instance, the intrusion kill chain (Hutchins et al., 2011) describes the stages of an attack, and MITRE ATT&CK framework (Alexander et al., 2020) presents the tactics and techniques that can be utilized in each stage of an attack. Both of these models focus on the attacker's actions, without explicitly outlining the target system. The Diamond Model takes a more detailed approach, by examining the capabilities and infrastructure of the attackers (Caltagirone et al., 2013).

The distinction between attack trees, and attack graphs (Lallie et al., 2020), can help us understand the difference between threat modeling and attack modeling. Although attack trees and attack graphs seem close to each other, as they use some graph notations, their practical usage is quite varied in terms of the main purpose.

Attack graphs are a more defensive approach to modeling attacks. They incorporate system knowledge, such as network access control rules and vulnerability lists, to identify possible attack paths to a victim system. Attack trees, on the other hand, are a more offensive approach to modeling attacks. They describe attack actions in a free-form text, without considering the system model. Although the experts can embed some knowledge about the target system and vulnerabilities to the nodes, this is totally up to those experts and done in an unsystematic way. Thus, an attack graph could be considered to be more closely aligned with threat modeling. Attack trees constitute the main method in various attack modeling studies and are also utilized as a complementary construct to known threat modeling approaches (e.g., STRIDE), as shown by Ahn et al. (2021).

2.4. STRIDE

The most mature threat modeling method, STRIDE, is an industry-developed threat modeling method. It was first introduced by Microsoft, as part of the Microsoft security development life cycle, *"to identify various types of threats the product is susceptible to during the design phase"*,[5] and has been used by Microsoft and other known organizations for decades.

STRIDE is a threat enumeration technique that helps to identify potential threats to a system. It is a mnemonic for **S**poofing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of service, and **E**levation of privilege. STRIDE analyzes the system components against 6 main security properties (*authenticity*, *integrity*, *non-repudiation*, *confidentiality*, *availability*, and *authorization*). This method is complemented by a system modeling approach, called a Data Flow Diagram (DFD). A DFD is a graphical representation of a system, and it is composed of four basic elements: processes, data stores, data flows, and external entities. DFD can be used to understand the system's architecture and what information is exchanged between the different components of a system.

STRIDE has two variants: STRIDE-per-Element and STRIDE-per-Interaction. STIRDE-per-Element focuses on identifying threats to each element of the DFD. STRIDE-per-Interaction enumerates threats based on the origin, destination, and interaction of data flows (Adam Shostack, 2013). Several studies have proposed combining STRIDE with other known security models. For example, a study by Ahn et al. (2021) proposes combining STRIDE with attack trees, which helps in better understanding the threats to a system.

2.5. TM application phase of the SDL

Threat modeling was first proposed to be applied in the early stages of the Software Development Life Cycle (SDLC), as one of the constructs of applying the security-by-design principles. As mentioned in Section 2.4, STRIDE was proposed as part of Microsoft SDL, which should occur in parallel to the SDLC. Microsoft SDL[6] starts with a prerequirement of training, followed by five phases: requirements, design, implementation, verification, and release phases. Microsoft SDL proposes performing TM at the design phase. However, some research has proposed the use of TM in other phases as well.

For example, Marksteiner et al. (2019) propose the use of threat modeling in automating the testing process for the system. The use of threat modeling techniques can also be extended to other phases in a system life cycle, such as the verification, release, and operation phases. For instance, assessing the system security, planning penetration testing

[5] https://www.microsoft.com/security/blog/2009/08/27/the-threats-to-our-products/.

[6] https://www.microsoft.com/en-us/download/details.aspx?id=12379.

(Salzillo et al., 2020; Rak et al., 2020), or prioritizing the system patching could benefit from TM. PatrIoT is a penetration testing methodology based on TM (Süren et al., 2023). TM is also promoted as a continuous process that should be considered in every design modification, not a one-time effort.

2.6. Privacy threat modeling

The privacy concept was introduced in an academic paper published in 1890. The paper defines privacy as *"the right to be let alone"* (Goldsmith et al., 1890). Due to its value, privacy has also been identified as one of the human rights acts.[7] There are many regulations related to data protection and privacy, including the ones related to specific sectors such as the Health Insurance Portability and Accountability Act (HIPAA),[8] and the Payment Card Industry Data Security Standard (PCI-DSS).[9] The General Data Protection Regulation (GDPR)[10] was put into effect in 2018, imposing obligations to protect the privacy of EU citizens.

As some systems, including ICSs, may store personal data, there is a significant need for threat modeling approaches that cover privacy aspects. LINDDUN is an acronym for the terms **L**inkability, **I**dentifiability, **N**on-repudiation, **D**etectability, information **D**isclosure, content **U**nawareness, policy and consent **N**on-compliance (Deng et al., 2011). The method shares a similar approach to STRIDE and was introduced as a framework to model privacy-specific threats for software-based systems. This framework categorizes privacy properties into *hard* and *soft* privacy. Hard privacy aims to conduct data minimization to reduce the need to trust the data controller, whereas soft privacy addresses data security countermeasures assuming that the data subject trusts the data controller.

Restricted Misuse Case Modeling (RMCM) is another threat modeling method that addresses the security and privacy requirements and was introduced in an industrial healthcare software case study (Mai et al., 2018). RMCM adopts a method named "RUCM" (Restricted Use Case Modeling) and extends it to meet security and privacy requirements.

2.7. Safety versus cyber-security in ICSs

In the past, only accidental component failures (hazards) were considered in the design of ICSs. However, the increasing use of communication technologies and open protocols, such as TCP/IP, have created more vulnerable systems to attacks launched in cyberspace. Cyber-attacks can cause incidents that may induce physical consequences, including human losses and property damages. Therefore, the interdependencies between the security and safety aspects should be considered during the threat analysis of ICSs (Kriaa et al., 2015).

The main difference between safety and security is that safety focuses on hazardous actions that occur unintentionally, whereas security threats are caused by a threat actor (sometimes called a threat agent) with malicious intentions. Threat modeling might consider both safety and security aspects in the system analysis. For instance, Suo et al. (2018) provide a STRIDE-based threat modeling method that was applied to an automated-driving system, and analyzes both safety and cyber security aspects.

It is worth noting that the European Parliament recently proposed a machinery regulation[11] that ensures the safety of machinery and robots. The proposed regulation states a difference between cybersecurity (a broad term related to networks) and cyber safety (related to machines).

2.8. Known threat modeling tools

There exist free and commercial threat modeling tools that help in automating the threat elicitation process. Free TM tools include Microsoft Threat Modeling Tool, OWASP Threat Dragon,[12] and Cairis.[13] While commercial TM tools include IriusRisk,[14] Tutamen Threat Model Automator,[15] SecuriCAD,[16] and Threatmodeler.[17] Commercial tools mostly support a community option that allows a single threat modeling trial.

The mentioned tools were designed mainly for software threat modeling, while some of them have expanded their scope to cover IoT and Cyber-Physical Systems (CPSs) as well. For instance, Microsoft Threat Modeling Tool provides the option to create customized threat modeling templates, which could be used to add different ICSs templates. The tool also supports a *"Medical Device Model"* template[18] that includes the representations of some cyber-physical devices related to the medical sector, similar to sensors and ionizing radiation source devices.

2.9. Threat modeling gamification

TM is an interdisciplinary process, that might be complex for persons with limited cyber security background. Tarandach and Coles (2020) list some threat modeling card games that could be used as an educational tool for teaching and practicing threat modeling. Some of the tools are based on STRIDE, similar to Elevation of Privilege[19] and Elevation of Privacy.[20] Other games do not follow a known threat modeling methodology similar to OWASP Cornucopia[21] and Security and Privacy Threat Discovery Cards.[22] The LINDDUN Team has created a card game called LINDDUN GO,[23] which is a lightweight version of LINDDUN that could be used by beginners.

2.10. Threat modeling in practice

Interviews with security experts who practice CPSs threat modeling in real life have shown that there is a need to enhance the relevant threat modeling methods (Jamil et al., 2022). According to the study, practitioners use a combination of threat modeling methods, approaches, and standards for CPSs threat modeling. However, this suggests that there is a lack of practical and systematic threat modeling methods for CPSs that suit the practitioners' needs. Also, this indicates a gap between threat modeling practices and academic research, as there are many models offered in the literature, while practitioners might not be fully aware of them.

Bernsmed and Jaatun (2019) discuss threat modeling challenges and best practices based on semi-structured interviews with experts that conduct threat modeling in agile software development for four different organizations. The interviews show that threat modeling is a time-consuming process, it is difficult to identify the relevant threats, software developers do not have the motivation to assign time for security, and the experts are not clear about when to stop the modeling process.

Soares Cruzes et al. (2018) identified 21 challenges faced by an organization while applying STRIDE within five agile development projects,

[7] https://www.echr.coe.int/documents/guide_art_8_eng.pdf.
[8] https://www.hhs.gov/hipaa/index.html.
[9] https://www.pcisecuritystandards.org/pci_security/.
[10] https://gdpr.eu/tag/gdpr/.
[11] https://ec.europa.eu/commission/presscorner/detail/en/ip_22_7741.
[12] https://owasp.org/www-project-threat-dragon/.
[13] https://cairis.org/.
[14] https://www.iriusrisk.com/threat-modeling-platform.
[15] https://www.tutamantic.com/page/features.
[16] https://foreseeti.com/securicad-enterprise/.
[17] https://threatmodeler.com/.
[18] https://github.com/microsoft/threat-modeling-templates.
[19] https://www.microsoft.com/en-us/download/details.aspx?id=20303.
[20] https://github.com/F-Secure/elevation-of-privacy.
[21] https://owasp.org/www-project-cornucopia/.
[22] https://securitycards.cs.washington.edu.
[23] https://www.linddun.org/go.

and also mapped the challenges found during the study to what was previously found in the literature. These challenges include challenges related to the threat modeling process, similar to the difficulty of drawing data flow diagrams, requiring security experts to run the threat modeling meetings, deciding when the analysis process is enough to stop, and engaging everyone in the discussions. The research team has also observed a decrease in motivation for security activities at the late threat modeling stages, similar to the threat identification stage. The research reasons this behavior by mentioning that in the initial asset identification stage meetings, the team was motivated for learning. However, while they advanced in the project to reach the threat identification stage, the meetings were oriented to document the threats, so they were not very engaging.

The threat modeling process might include the use of a threat library or attack taxonomy to enhance the efficiency of the process. While it is argued that threat libraries limit the possibility of discovering new threats, as the findings might be limited to special architectures and cannot be applied to different system architectures (Uzunov and Fernandez, 2014). On the other side, threat taxonomies are mostly at a high abstract level and require good security knowledge to identify relevant system threats. Through this literature review discussion section, we also discuss the TM challenges noticed in the selected studies that addressed ICSs TM.

3. Related work

To our knowledge, this is the first systematic literature review on threat modeling that focus on ICSs and analyzes the studies from the control systems perspective. Other systematic literature reviews have addressed TM in a more general context. Tuma et al. (2018) SLR focuses on threat modeling of software systems and analyze 25 threat modeling methodologies. The paper's main findings are: the analysis procedures are not precisely defined, there is a lack of quality assurance of the analysis outcomes and tool support, and validation is limited. Another SLR about threat modeling is presented by Xiong and Lagerström (2019), without limitation to a specific application domain. The review spots that threat modeling definitions are various and the domain is missing common ground.

The SLR of Ling et al. (2020) concludes six sources of information to be used in power systems threat modeling as follows: expert knowledge, logs and alerts, previous research, system's state, vulnerability scoring and databases, and vulnerability scanners. This paper reviews only the information sources that can be used for TMs but does not provide an in-depth discussion about the TM methods.

Luo et al. (2021) present an SLR of Threat Analysis and Risk Assessment (TARA) for Connected Vehicles. Kriaa et al. (2015) present a survey related to risk assessment methods that combine both safety and security aspects in ICSs. The paper does not use the *"threat modeling"* terminology while referring to many standards and risk assessment methods that focus on interdependencies between safety and security in ICSs. This survey is relatively old (i.e., published in 2015), which means recent papers regarding ICSs were not covered.

Nweke and Wolthusen (2020) review four asset-centric threat modeling approaches for software design (DREAD, Trike, OCTAVE, and PASTA). Shevchenko et al. (2018a) summarize twelve methods of threat modeling (STRIDE and its Derivations, PASTA, LINDDUN, CVSS, Attack Trees, Persona non Grata, Security Cards, hTMM, Quantitative Threat Modeling Method, Trike, VAST modeling, and OCTAVE). The study includes methods similar to CVSS and attack trees, which we do not agree on categorizing as threat modeling methods by themselves, while they might be used within threat modeling methods. Shevchenko et al. (2018b) continue the work done in Shevchenko et al. (2018a) and evaluate the twelve threat modeling methods based on five criteria: 1) Strengths and weaknesses (maturity and usage, focus/perspective, time to implement, effectiveness and mitigation strategies), 2) Adoptability (easy to use, easy to learn, and documentation and support), 3) Tailorability (integration with SDLC, compatibility with the agile development process, and scalability), 4) Applicability (coverage of safety-security interdependency, and integration of hardware and software threats), and 5) Automation (availability of tools, integration options for tools into an SDLC, portability of tools). However, the evaluation criteria for each method do not follow an approach with special metrics. For example, the evaluation of the "time/effort" aspects, which should reflect how the method is time-consuming, does not have clear criteria on how Shevchenko et al. (2018a) reached the mentioned results, the same applied to other aspects similar to "consistent results". Shevchenko et al. (2018a) recommend using the PASTA threat modeling method due to its detailed guidance through the modeling process and its flexibility. The research also proposes some additional steps while applying PASTA on a CPS, to cover more aspects related to security, safety, and privacy. However, the study did not demonstrate its effectiveness by applying it to a case study, which leaves the readers with unanswered questions about the procedures of applying PASTA to CPSs.

The previous review papers do not provide a clear and detailed overview of the proposed ICSs threat modeling methods in the literature. In addition to specific analysis within the ICSs context, this literature review provides detailed background about the field, puts special emphasis on threat modeling stages, and proposes a taxonomy to categorize ICSs threat modeling methods.

4. Research methodology

To answer the research questions discussed in Section 1, we followed the guidelines for systematic literature review proposed by Kitchenham et al. (2007), as shown in Fig. 1.

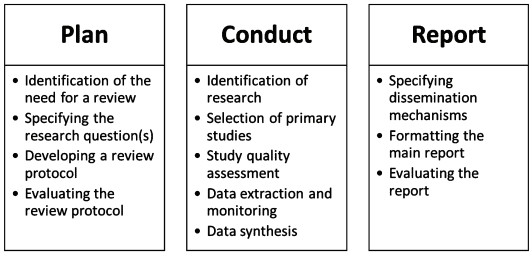


Fig. 1. SLR stages proposed by Kitchenham et al. (2007).

4.1. Planning the review

This SLR was initiated in 2021, as our team was eager to understand the state of the art of ICSs TM. We first reviewed the related studies discussed in Section 3. However, these studies did not provide us with a clear understanding of the current TM methods that are used in ICSs, while considering the unique nature of such CPSs.

As our team was involved in the secure design process of a power system, it was important for us to find a well-documented, simple, and practical ICSs TM methodology that we could adapt to our real-world project. For this reason, we created an ICSs TM taxonomy and clearly described the list of TM stages that we identified in the literature. We believe that this categorization could help the readers to better understand the strengths and weaknesses of each of the selected ICSs TM approaches. Although the categorization of threat modeling approaches based on taxonomy and applied stages is not a standard practice in systematic literature reviews, we believe that it is a valuable contribution to this study. Therefore, we have separated the categorization criteria in a separate section, Section 5.

We also decided to include an extensive and comprehensive background to this SLR, where we introduce to the reader different TM

concepts, as we found a clear gap in knowledge of these concepts in other similar SLRs. This SLR was written over a period of two years, during which we performed TM in practice in parallel. Such experience greatly enhanced the research quality and helped us to better understand the identified gaps in the current literature.

We have iteratively modified and enhanced the review protocol throughout the research process to improve the completeness of the results. Due to resource constraints, the review protocol was discussed with the second author and continuously enhanced based on trial results. For example, the initial search for this SLR was conducted using the search criteria:("threat model" AND ("ICS" OR "industrial control system" OR "SCADA" OR "supervisory control and data acquisition" OR "distributed control system" OR "DCS")). The asterisk (*) wildcard was used to expand the keywords *model*, and *system* to include different variations of these keywords, such as *threat modeling* and *threat modelling*. This helped us to identify a broader range of papers that were relevant to our research. However, this search criteria returned a limited number of studies (as of May 2021, it returned 37 papers, of which only 11 were kept for full reading). Therefore, during the same month, we iteratively expanded the research keywords to include the *cyber physical, cyber-physical* and *CPS* keywords, which resulted in 44 papers for full reading. This confirms that adding the CPS-related keywords helped us to identify more studies that are relevant to our review. Fig. 2 summarizes the final version of the review protocol used for this SLR.

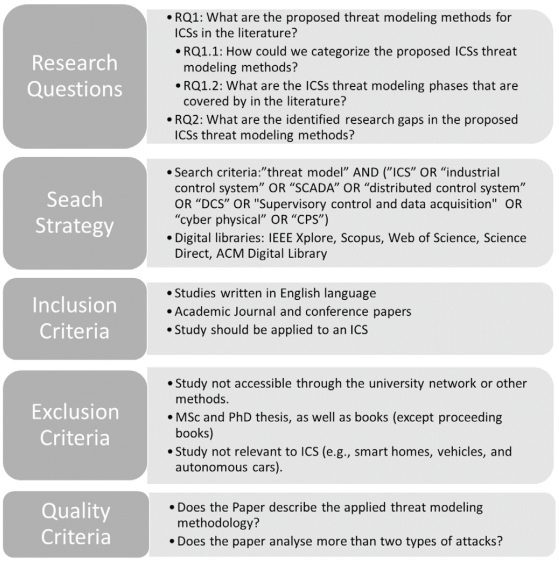


Fig. 2. SLR Search protocol.

4.2. Conducting the review

The first round of searches was conducted on five selected search libraries in May 2021, namely IEEE Xplore, Scopus, Web of Science, Science Direct, and ACM Digital Library. The scope of the search included the title, abstract, and keywords. This search returned 44 papers for full reading. The search was repeated in February 2022, which resulted in an additional 8 papers for full reading. A third search was conducted in June 2023, which resulted in another 32 papers for full reading. This made a total of 84 papers that were fully read through the search criteria mentioned in Fig. 2.

4.2.1. Selection process

On June 30, 2023, we conducted a third search of the five selected search libraries. This search returned a total of 288 papers, as shown in Fig. 3. After eliminating duplicate papers, we were left with 162 papers. We then compared this list with the lists from the previous search iterations and found that seven studies were missing. We manually added these studies to the list, bringing the total number of papers to 169. However, we were only able to access 162 studies.

We proceeded to read the abstracts of the 162 accessible papers. We had already read the abstracts of 105 of these papers in previous search iterations, so we mainly read 57 new abstracts. We used the inclusion-exclusion criteria to exclude papers that did not match our inclusion criteria from an abstract read. We conducted full paper readings for the papers that remained after reading the abstracts.

We were left with 106 papers for full paper reading, of which 84 came from the search criteria and 22 came from snowballing, as well as looking into other threat modeling research groups' publications. Fig. 3 shows the results of the search criteria, as well as the number of identified papers through the different filtration stages. We selected 36 papers that met the inclusion, exclusion, and quality criteria. Of these, 27 papers were found through the systematic literature review search, and 9 papers were found through snowballing.

4.2.2. Exclusion, inclusion, and quality criteria

This SLR excluded papers that do not allow access through the network of Tallinn University of Technology and that we could not access using other methods, such as contacting the authors. We also excluded master's and PhD theses, as well as irrelevant papers to ICSs such as those related to smart homes, vehicles, and autonomous cars as application areas for threat modeling.

To ensure that the papers included in the research propose a systematic ICSs TM methodology that the readers can apply, the quality criteria questions, mentioned in Fig. 2, had to be answered by "yes" in order to include the research in this SLR analysis. Papers discussing general information about security concerns in ICSs, without proposing a threat modeling methodology, were excluded. Additionally, papers that analyzed only one or two types of attacks were excluded because threat modeling, when applied during the design phase, should strive for completeness by considering a wide range of potential threats.

This research includes threat modeling academic papers that are written in English and focus on ICSs. The included papers analyze systems with different control components, such as Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs), and Intelligent Electronic Devices (IEDs). They also may have sensors, actuators, and other similar components that might be found in a typical ICS. The included papers discuss at least three different types of threats (e.g., False Data Injection (FDI), Man in the Middle (MITM), and Denial of Service (DOS) threats) to provide a threshold for completeness of the threat analysis. The papers included in this research are either the ones that apply or propose a threat modeling methodology, a TM Language, or a tool for automating the threat modeling process.

5. Categorization criteria

The selected 36 papers were categorized based on a proposed threat modeling taxonomy; then, each study was mapped to the threat modeling phases that were applied by it. This section introduces the categorization criteria.

5.1. ICSs threat modeling taxonomy

Based on the literature, we identified various aspects to categorize the ICSs threat modeling studies and aligned them in a descriptive taxonomy, as shown in Fig. 4. The taxonomy categorizes papers based on their *method type, approach, focus, validation method*, and *application area*. Each of these aspects is explained in the following sub-sections.

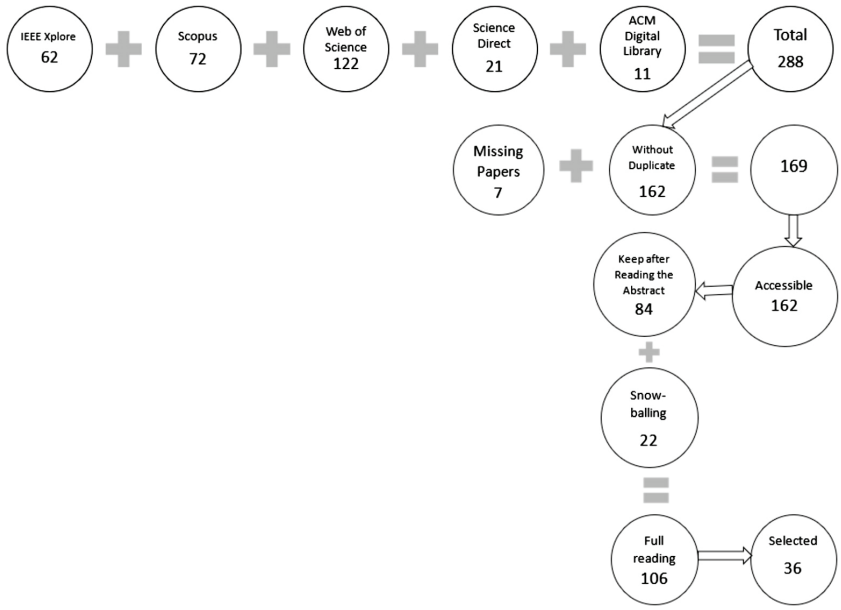


Fig. 3. Number of studies through the selection process.

5.1.1. Method type

As threat modeling is a broad terminology, it might be done differently, depending on the stakeholders' needs. We divided the TM methods types into five subcategories: *formal, semi-formal*, and *non-formal* methods, *modeling language*, and *tool*.

Formal method Formal threat modeling methods propose a TM approach that is fully based on mathematics. However, it is challenging to find a formal threat model that has a complete representation of CPSs threats, as considering software, hardware, and physical threats simultaneously is a challenge (Burmester et al., 2012). Such methods require specific mathematical knowledge, which makes them more suitable for academic research, while they are still relatively unpopular in practice. Formal methods could also be used to create threat modeling software that automates the TM process.

Non-formal method This category does not resort to formal representations. Non-formal TM methods mostly depend on the involvement of experts in every stage of the threat modeling process. For example, the experts create the ICSs system mapping using data flow diagrams, attack trees, or Petri-nets. In non-formal methods, the threat enumeration is done by the experts, and they decide how to prioritize the threats. They choose the prioritization criteria based on their needs and knowledge, without any advanced tool support or mathematical calculations. Such methods tend to analyze several types of threats (e.g., STRIDE analyzes 6 types of threats), and they are widely used in organizations, as they are easy to understand by different stakeholders.

Semi-formal method This SLR categorizes semi-formal TM methods as methods that combine formal and non-formal modeling constructs in a single method. In semi-formal methods, threats might be elicited by the team using a non-formal method (e.g., STRIDE), then the risk might be calculated using a formal method for calculating the risk of the identified threats. Within the context of this SLR, we considered combining non-formal methods with a Matlab simulation as a semi-formal method, as such simulation is based on a mathematical representation of the system.

Modeling language Threat modeling languages are mostly domain-specific modeling languages (DSLs) that map system components, their interactions, security constraints, requirements, or threats using a specific language, such as UMLsec or SecureUML (Johnson et al., 2018). DSLs can be used as a basis for creating TM tools.

Tool Threat modeling tools aim to automate the ICSs TM process. Examples of TM tools were discussed in Section 2.8.

5.1.2. Approaches

In general, threat modeling has three widely known approaches, *asset-centric, attack(er)-centric*, and *system-centric* (also named software-centric). The literature includes a fourth approach, the *data-centric* approach, which focuses on *"protecting particular types of data within systems"* (Souppaya and Scarfone, 2016). However, the proposed ICSs TM taxonomy excludes the data-centric threat modeling approach, as from its description, it does not seem to analyze the impact of threats on the physical system, which makes it more suitable for IT systems, not ICSs.

Asset-centric This approach only focuses on valuable assets during the threat analysis. As per Adam Shostack (2013), assets could be described in 3 ways: *a "thing" targeted by attackers, a "thing" that needs to be protected, or a "stepping stone" for these two*. The term asset does not only refer to system components (tangible assets), but it also includes business assets (intangible assets), such as the organization's image or reputation. Asset-centric threat modeling gives a practical perspective for the TM, as it also studies the impact of losses on the business. In this approach, critical assets are first identified, then mapped in a diagram to show interconnections between the different assets (Adam Shostack, 2013).

DREAD (Howard and LeBlanc, 2002), Trike (Saitta et al., 2005), OCTAVE (Alberts et al., 2003), and PASTA (UcedaVelez and Morana, 2015) are well-known asset-centric threat modeling methods. The review of these four threat modeling approaches, provided by Nweke and Wolthusen (2020), shows that they all contribute to risk management and adds that an asset-centric approach is sometimes referred to as a risk-based threat modeling approach. From the literature, we noticed

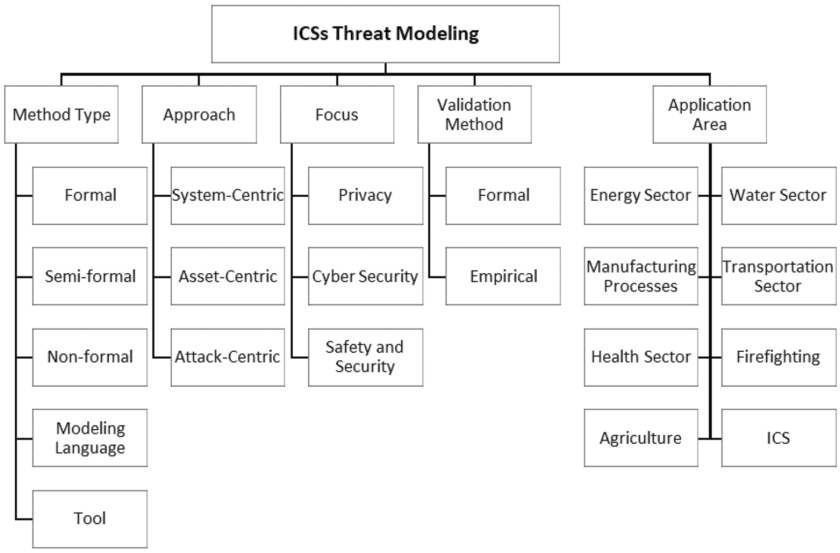


Fig. 4. Proposed ICSs threat modeling taxonomy.

that asset-centric threat modeling methods are sometimes referred to as risk assessment methods. The term *threat modeling* may not be used in papers that discuss this approach.

Attack(er)-centric This threat modeling approach focuses on the attacker's perspective. The description of this approach varies in the literature. Adam Shostack (2013) describes it as an approach that identifies the attacker types, similar to "personas" for threat agents. On the other hand, Schlegel et al. (2015) mentions that this approach also describes how the attacker would get into the system. We also noticed that Withers (2016) use both terms attack-centric and attacker-centric to describe this approach.

In general, the attack(er)–centric approach tends to identify the types of attackers that may target the system, such as state-funded groups or cyber-criminals. It also explores their objectives, resources, capabilities, and other aspects that might impact the attack results. For example, an attacker's objectives could be to steal data, disrupt operations, or damage the system. Their resources could include financial resources. Their capabilities could include the ability to exploit known vulnerabilities or perform complex zero-day attacks. However, such an approach might also use attack graphs (Withers, 2016), or attack trees (Paverd et al., 2014) to characterize threats. Zografopoulos et al. (2021a) propose a TM method that can be categorised as attack(er)-centric. The proposed threat model combines the adversary model (knowledge, access, specificity, and resources) and the attack model (frequency, reproducibility, discoverability, functional level, asset techniques, and premise). The attacker-centric approach was also referenced as security-centric by Morana and UcedaVelez (2015).

System-centric As threat modeling methodologies were first introduced to secure software through the development life cycle, this approach is also known as *software-centric* approach. However, we find the term *system-centric* approach to be more relevant to ICSs threat modeling. The idea of using a system-centric approach is to map out the whole system in diagrams and study the threats through each of the system components. This approach requires a better understanding of the functions of the entire system, which might be a challenging task for complex systems. However, when system designers are involved in the threat modeling process, the system-centric approach seems to be easy to adapt, as it does not require good knowledge of the business assets or attacker's profiles as the two other approaches (Adam Shostack, 2013).

It is interesting to mention that STRIDE is categorized as attacker-centric (Morana and UcedaVelez, 2015), and system-centric (Hajrić et al., 2020; Withers, 2016; Adam Shostack, 2013) TM approaches. However, we tend to categorize it as a system-centric threat modeling approach, as it maps the whole system and analyzes its threats. Different TM approaches might be combined to provide a "complete" list of threats (Jamil et al., 2022). For example, the system-centric approach was combined with the attacker-centric by Withers (2016).

5.1.3. Threat model focus

This SLR categorizes the focus of an ICSs threat model as privacy focused, cyber security focused, and safety and security focused.

Privacy focused TM A study is categorized as privacy focused if its primary objective is to explore privacy-related threats.

Cyber security focused TM Widely used threat modeling methods (e.g., STRIDE) analyze cyber-security aspects, similar to the CIA triad terminology (Confidentiality, Integrity, and Availability), which is derived from the classical view of information-centric loss in IT systems.

Safety and security focused TM An ICS carries the properties of a CPS, which requires a TM method that combines both safety and security aspects into its analysis. If a study has the sole focus on safety, we excluded it from our review as it is beyond the scope of this SLR.

5.1.4. Threat model validation method

Validation of a proposed threat modeling method might be done through a formal method or using an empirical validation method. Empirical validation is done based on qualitative or/and quantitative research.

5.1.5. Threat model application area

This SLR identifies the TM application areas for ICSs as the energy sector (i.e., power, oil, and gas), water, manufacturing process, transportation, and health. During the third iteration of this study, we noticed new application areas of ICSs TM, such as agriculture and

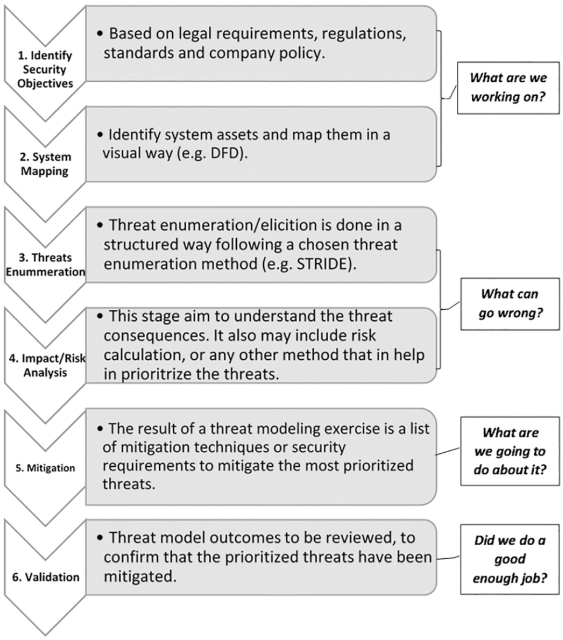


Fig. 5. Identified threat modeling stages.

firefighting. The term ICS is used as an application area in case the categorized paper does not indicate a specific application area.

5.2. Threat modeling stages

The literature shows that there is no unique and systematic threat modeling process to follow, as the process steps vary from one model to another. For example, Microsoft first defined threat modeling in a four-step process: *diagramming*, *threat enumeration*, *mitigation*, and *verification* (Shostack, 2008). Then they updated the threat model phases to 5 steps, after adding the *definition of security requirements*, as the first phase.[24] On the other hand, Steven (2010) proposes eight major steps for threat modeling and each step includes sub-steps: (1) diagram the system, (2) identify assets, (3) identify threats, (4) marry threats and software, (4) enumerate potential "doomsday" impact scenarios, (5) document specific misuse or abuses, (7) enumerate attack vectors, (8) output items (from 6.1 and 7.3) to validate via later application assessment activities.

Based on the literature, as well as the discussed definitions in Section 2.1, we identified six stages of a threat modeling process, as shown in Fig. 5. The first stage is *Identify Security Objectives* based on legal requirements, regulations, standards, and organizational policy. *Identify Security Goals* might also be included in this stage. Examples of identified security goals might be data confidentiality, integrity, and availability, which are aspects related to IT systems. However, as our analysis scope is ICSs, we may also identify the OT-related goals at this stage (e.g., safety, reliability, and availability). The aim of setting security objectives/goals at this stage is to facilitate the security requirement elicitation later and provide a justification for requirements selection (Yu and Mylopoulos, 1998).

In the second stage, *System Mapping*, system components are identified, and data flows are mapped. The first two stages aim to answer the question, "what are we working on?". The question "what can go wrong" is answered using the third stage, *Threat Enumeration*, where threats are identified. In some models, threat enumeration is followed by an impact assessment and/or risk calculation, to help in threat prioritization. We find that this *Impact/Risk analysis* stage is a fundamental stage for ICSs, as it helps to understand the consequences the threats and ensure the prioritization of critical threats mitigation. For this reason, we consider that the 4th stage also answers the question "What can go wrong?".

The fifth stage, *Mitigation*, should propose mitigation techniques for the identified threats, as this answers the question "What are we going to do about it?". In case the threat modeling process was run in the design phase, then the outcome of this stage might be a list of security requirements or security controls.

The last stage, Validation, aims to answer the question "Did we do a good enough job?". This stage should include validation for the threat modeling analysis, to make sure that all the addressed threats were effectively mitigated. This stage could be performed through peer review, TM tool, or any other method the team agrees on. We must mention that the described six stages seem to suit non-formal ICSs methods better, while it might be challenging to categorize non/semi-formal methods and tools using these stages.

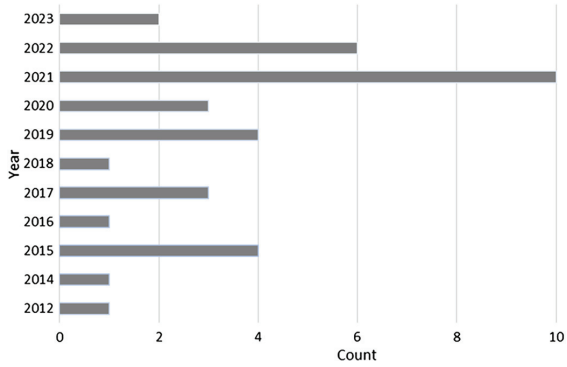


Fig. 6. The number of selected ICSs threat modeling papers per year of publication.

6. Results

This section discusses the results of the analysis of the selected papers. The analysis includes general information about the papers, such as publication year and the first author's institute location. The papers were also analyzed based on the proposed ICSs TM taxonomy, as well as the covered threat modeling phases.

6.1. General information

The analysis shown in this section is based on the selected papers that propose/apply TM for ICSs. Fig. 6 shows the number of published ICSs threat modeling papers per year of publication. The oldest selected paper is dated to 2012, which we relate to the reveal of Stuxnet in 2010, as this incident was a major motive for researchers and practitioners to prioritize ICSs security. We notice a peak in the number of ICSs TM papers during 2021, which we might refer to the growing number of attacks on ICSs, especially power grids, resulting in more research in this field. The COVID-19 pandemic may have led to a higher number of high-quality ICSs TM publications in 2021, as researchers focused on research during this time.

When looking into the location of the first author's institutions, it was found that 18 papers (50%), out of the selected 36, were written by a first author's institute located in Europe, 6 located in North America (17%), and 12 located in Asia (33%), as shown in Fig. 7. When

[24] https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling.

Table 1
Details of papers cited more than 100 times.

Reference.	Publication Venue	Citations
Friedberg et al. (2017)	Journal of Information Security and Applications	243
Khan et al. (2017)	2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)	198
Liu et al. (2015)	IEEE Transactions on Smart Grid	168
Zografopoulos et al. (2021a)	IEEE Access	134

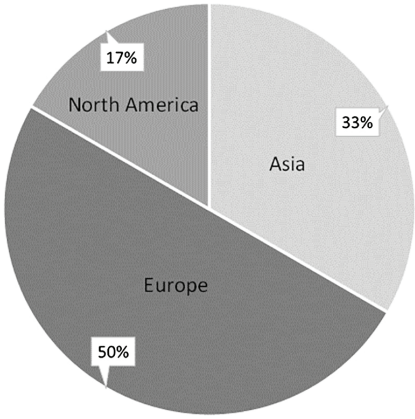


Fig. 7. The location of the first authors' institutes of the selected ICSs TM papers.

looking into the number of authors for each paper, we found that one paper is written by a single author, Fernandez (2016), while the rest of the papers are written by two or more authors. Some papers show the contribution between different institutes located in the same continent (e.g., Friedberg et al. (2017), Radoglou-Grammatikis et al. (2019)), or different continents (e.g., Ahn et al. (2021), Suleiman et al. (2015)).

We analyzed the venues where the selected papers were published. Out of the 36 selected papers, 18 papers are conference papers, 12 of which are published by IEEE conferences. The selected papers include 18 journal papers, where 9 papers are published in Elsevier journals (Journal of Systems and Software, Information Systems, Journal of Information Security and Applications, International Journal of Critical Infrastructure Protection, Internet of Things, Computers in Industry, Journal of Systems Architecture, and Computers & Security), and 4 papers are published in IEEE journals (IEEE Transactions on Smart Grid, IEEE Transactions on Dependable and Secure Computing, and IEEE Access).

As per Google Scholar on August 2023, four selected papers (three Journal papers and a conference paper) has been cited more than 100 times, as shown in Table 1. The most cited paper, Friedberg et al. (2017), proposes a TM method that integrates safety analysis (STPA) and security analysis (STPA-sec) into one framework, STPA-SafeSec, and includes the analysis of control loops in the proposed method. The STPA-SafeSec paper was cited 243 times. The second most cited paper, Khan et al. (2017), applies STRIDE on a Synchronous islanding testbed, which is related to the power industry applications, and the paper was cited 198 times. Liu et al. (2015) main focus is not threat modeling, while the study used threat modeling as a stage toward the creation of an intrusion detection system for Advanced Metering Infrastructure (AMI) and the paper was cited 168 times. Zografopoulos et al. (2021a) apply an attack-centric threat model to an energy system and was cited 134 times.

6.2. Analysis based on the ICSs TM taxonomy

Table 2 summarizes the analysis of the selected ICSs threat modeling papers based on the proposed TM taxonomy, discussed in Section 5. The analysis shows that the proposed TM methods mostly involve a system-centric approach in their threat analysis (23 papers), which focuses on the whole system's threats. We also noticed that non-formal methods tend to use a system-centric approach, as 15 out of 19 non-formal papers adopt a system-centric approach. On the other side, four out of five modeling language papers were found to use an attack-centric approach. The attack-centric approach is combined with the Asset-centric approach by Stellios et al. (2021); Lee et al. (2021); Jbair et al. (2022) and combined with the system-centric approach by Suleiman and Svetinovic (2013); Ahn et al. (2021). We also noticed that the asset-centric studies Stellios et al. (2021); Rimsha and Rimsha (2019); Zografopoulos et al. (2021b) perform risk calculation for critical assets. The third search iteration, done in June 2023, included the study by Lee et al. (2021), which uses a data-centric approach combined with asset and attack-centric approaches. The study used STRIDE for threat enumeration, while the enumerated threats were only threats related to data flow.

Of the 36 selected papers, 25 focus solely on cyber security, while a single paper, Wuyts et al. (2014), focuses solely on privacy. Two papers focus on both privacy and cyber security (Chen and Huang, 2019; Zahid et al., 2023). Six papers focus on safety and security (without considering privacy) which supports the comprehensiveness of the TM results, as ICSs safety is fundamental, and it could be impacted by cyber-security threats. For instance, STPA-SafeSec (Friedberg et al., 2017), investigates the interrelationship between safety and security vulnerabilities. The study combines safety analysis methods, STPA, and STPA-sec into a single approach to balance both safety and security constraints. We have considered studies that identify the threat consequences, such as Khan et al. (2017), and Khalil et al. (2023), to focus on both ICSs security and safety, as they consider the physical threat consequences that may impact the system safety and cause human injuries, or system damage.

Out of the 36 selected papers, only Two papers consider threats related to safety, security, and privacy, Suleiman et al. (2015) and Amro et al. (2023). The paper by Suleiman et al. (2015) elicits threats based on SQUARE and SREP methods. However, it does not clarify the procedures of the mentioned methods. The research uses a Smart Grid (SG) system as a case study and elicited 76 related threats that are categorized into 6 categories: threats to network availability, threats to data integrity, threats to data confidentiality and information privacy, threats to SG key operation, threats to inter-organizational collaborations and relationships, and threats to the resilience and safety of SG infrastructure. The study by Amro et al. (2023) uses a bottom-up risk assessment approach called *Failure Modes Effects and Criticality Analysis* (FMECA). The authors identify the failure modes of autonomous passenger ships, based on the MITRE ATT&CK framework. The paper provides extensive methodology details, which might help the reader to adapt the proposed method in TM of other application areas of ICSs.

The analysis of selected papers shows that 35 used methods are validated using empirical validation. Whereas 34 papers use qualitative methods for validation, and a single paper uses a quantitative research method. The qualitative validation methods are mostly done through use cases and case studies, except the study of Suleiman and Svetinovic (2013), who did further validation for the used ICSs TM method based

Table 2
Categorization of the selected ICSs TM papers based on the proposed ICSs TM taxonomy.

Reference	Method Type					Approach			Focus			Validation		Application Area
	Tool	Mod. Lang.	Form.	Non-Form.	Semi-Form.	Asset-Centric	Attack-Centric	System-Centric	Priv.	Safety + Sec.	Cyber Sec.	Form.	Emp.	
Wuyts et al. (2014)				Y				Y	Y				Y	Energy
Suleiman et al. (2015)				Y				Y	Y	Y			Y	Energy
Suleiman and Svetinovic (2013)				Y			Y	Y			Y		Y	Energy
Chen and Huang (2019)					Y			Y	Y		Y		Y	Energy
Ramis Ferrer et al. (2017)				Y				Y			Y		Y	Manufact.
Ahn et al. (2021)					Y		Y	Y			Y		Y	Energy
Friedberg et al. (2017)				Y				Y		Y			Y	Energy
Khan et al. (2017)				Y				Y		Y			Y	Energy
Fernandez (2016)		Y					Y			Y			Y	Transport
Haider et al. (2019)				Y				Y			Y		Y	Energy
Radoglou-Grammatikis et al. (2019)				Y				Y			Y		Y	ICS
Zografopoulos et al. (2021a)					Y		Y				Y		Y	Energy
Stellios et al. (2021)					Y	Y	Y				Y		Y	Health
Girdhar et al. (2021)				Y				Y			Y		Y	Energy
Fla et al. (2021)	Y							Y			Y		Y	Energy
Li et al. (2021)				Y				Y			Y		Y	ICS
Sun et al. (2020)		Y					Y				Y		Y	Water
Rimsha and Rimsha (2019)	Y					Y					Y			Energy
Schlegel et al. (2015)	Y							Y			Y		Y	Energy
Martins et al. (2015)	Y							Y			Y		Y	Transport
Vernotte et al. (2018)	Y						Y				Y		Y	Energy
Hacks et al. (2020)		Y					Y				Y		Y	Energy
Zografopoulos et al. (2021b)				Y		Y					Y		Y	Energy
Liu et al. (2015)				Y				Y			Y		Y	Energy
Foldvari et al. (2022)	Y							Y		Y			Y	Water
Kumar et al. (2022)				Y			Y				Y		Y	Energy
Amro et al. (2023)					Y		Y		Y	Y			Y	Transport
Lee et al. (2021)				Y		Y	Y				Y		Y	ICS
Al Asif et al. (2021)				Y				Y			Y		Y	Agriculture
Kim et al. (2022)				Y				Y			Y		Y	Energy
Shibly and De Soto (2020)				Y				Y			Y		Y	Manufact.
Zahid et al. (2023)				Y			Y		Y		Y		Y	Firefight.
Khalil et al. (2023)				Y				Y		Y			Y	Energy
Valenza et al. (2022)	Y		Y					Y			Y		Y	Energy
Jbair et al. (2022)	Y	Y				Y	Y			Y			Y	Manufact.
Rouland et al. (2021)	Y	Y	Y					Y			Y		Y	Energy

on a list of identified evaluation criteria. Using a case study, Suleiman and Svetinovic (2013) evaluate the Security Quality Requirements Engineering (SQUARE) (Mead and Stehney, 2005) based on qualitative research that analyzes the method's: flexibility, sampling, analyst permissibility, interpretiveness, data sufficiency, coherence, repeatability, completeness, usability, and credibility. The evaluation concludes that SQUARE does not guarantee the completeness of security goals identification, also it is not clear when to stop each of the SQUARE steps.

The study presented by Wuyts et al. (2014) uses a quantitative empirical method to validate the LINDDUN TM method based on four aspects: correctness, completeness, productivity, and ease of use. The experiment setup of Wuyts et al. (2014) was also previously used by Scandariato et al. (2015) to validate STRIDE as a software TM method. The LINDDUN experiment included 3 studies for validation. Two descriptive studies worked on the requirement analysis and design phases, where students discover privacy threats and design weaknesses. The third study involved privacy experts, to compare their findings to the LINDDUN method findings. The research outcomes showed that LINDDUN was considered easy to learn and the students provided reasonable levels of correctness ratios. However, the method completeness could be improved and the threat tree catalogs could explain some threats better.

As for the application areas of the proposed ICSs TM methods, we have marked papers with an "ICS" sector when they do not focus on a specific application area, instead, they focus on ICSs in general. The energy sector is the most common application area for ICSs TM methods, with 22 studies focusing on this sector. This is likely due to the critical nature of the energy sector, as many sectors depend on it, and the potential for significant damage that could be caused by a cyberattack. The Ukrainian power grid incidents were a stark reminder of the need for better security of energy systems against cyber attacks. These incidents caused widespread power outages and highlighted how cyber threats could cause large damage in such a critical sector.

This review also includes transport-related papers. Fernandez (2016) focus on a port loading system, and Martins et al. (2015) focus on a real-world wireless railway temperature monitoring system. The water sector is covered by the studies Sun et al. (2020) and Foldvari et al. (2022). Ramis Ferrer et al. (2017) is applied to a manufacturing process, and Jbair et al. (2022) is also applied to smart manufacturing applications. Al Asif et al. (2021) focus on the agriculture sector, while Zahid et al. (2023) focus on a smart firefighting system. As general application areas of ICS, Radoglou-Grammatikis et al. (2019) use a SCADA testbed in their use case, while Li et al. (2021) use a PLC-related case study.

6.3. ICSs TM methods analysis based on the applied TM stages

The selected papers were analyzed based on the covered TM stages, as shown in Table 3. We noticed only three papers, Kim et al. (2022), Zahid et al. (2023) and Suleiman and Svetinovic (2013) mention the security objectives phase, while all the other studies start the threat modeling process from the system mapping phase. We could relate this behavior to the fact that most of the papers were not applying the proposed method to a real-world system, so they were not required to identify such objectives. Also, the use of methods similar to STRIDE and LINDDUN might be considered as a substitute for this first stage, as these methods already analyze the system against aspects that could be considered as security and privacy objectives. However, we have to stress that in a real-life scenario, identifying the security objectives is

Table 3
Covered TM phases by the selected papers.

Reference	Identify Security Objectives	System Mapping	Threat Enumeration	Impact/Risk Analysis	Mitigation	Validation
Wuyts et al. (2014)		Y	Y	Out of Scope	Out of Scope	
Suleiman et al. (2015)		Y	Y		Out of Scope	
Suleiman and Svetinovic (2013)	Y	Y	Y	Y	Y	Y
Chen and Huang (2019)		Y	Y	Y		
Ramis Ferrer et al. (2017)		Y	Y	Y	Y	
Ahn et al. (2021)		Y	Y	Y	Out of Scope	
Friedberg et al. (2017)		Y	Y	Y		
Khan et al. (2017)		Y	Y	Y	Out of Scope	
Fernandez (2016)		Y	Y		Y	
Haider et al. (2019)		Y	Y	Y	Y	
Radoglou-Grammatikis et al. (2019)		Y	Y	Y		
Zografopoulos et al. (2021a)		Y	Y	Y		
Stellios et al. (2021)		Y	Y	Y	Y	
Girdhar et al. (2021)		Y	Y	Y	Y	
Fla et al. (2021)		Y	Y			
Li et al. (2021)		Y	Y			
Sun et al. (2020)		Y	Y	Y		
Rimsha and Rimsha (2019)		Y	Y	Y		
Schlegel et al. (2015)		Y	Y	Y	Y	Y
Martins et al. (2015)		Y	Y	Y	Y	
Vernotte et al. (2018)		Y	Y		Y	
Hacks et al. (2020)		Y	Y	Y	Y	
Zografopoulos et al. (2021b)		Y	Y	Y		
Liu et al. (2015)		Y	Y		Y	
Foldvari et al. (2022)		Y	Y	Y		
Kumar et al. (2022)		Y	Y			
Amro et al. (2023)		Y	Y	Y	Y	
Lee et al. (2021)		Y	Y		Y	
Al Asif et al. (2021)		Y	Y		Y	
Kim et al. (2022)	Y	Y	Y	Y	Y	
Shibly and De Soto (2020)		Y	Y	Y	Y	
Zahid et al. (2023)	Y	Y	Y		Y	
Khalil et al. (2023)		Y	Y	Y	Y	Y
Valenza et al. (2022)		Y	Y	Y		
Jbair et al. (2022)		Y	Y	Y	Y	
Rouland et al. (2021)		Y	Y		Y	

a fundamental case, as ICSs have to meet the organizational policy, as well as industry standards and regulations.

It was also found that all the selected methods go through the process of system mapping and threat enumeration. As they are the fundamental stages of threat modeling. Impact and risk assessment are important activities for prioritizing threats and choosing suitable security requirements and controls to mitigate critical threats. The analysis shows that most of the papers consider the impact/risk analysis phase. When a stage is mentioned by the paper, but it is set out of the scope of the paper, then we mention it as *out of scope* in Table 3.

We noticed that most of the studies calculate the risk using impact and likelihood. However, some papers have different definitions of how to perform the impact or risk assessment. For example, Girdhar et al. (2021) mention that STRIDE is a *"categorical risk assessment model"* and that the risk analysis is done through the use of STRIDE. The paper does not calculate the risk using impact and likelihood, as it is commonly done, instead, during the threat elicitation phase, it mentions what we may consider as the "impact" of each threat. Also, we marked that Suleiman and Svetinovic (2013) perform impact/risk analysis, while it is interesting to mention that this method performs the prioritization after eliciting the security requirements instead of prioritizing the elicited threats. Khalil et al. (2023) map losses to threat consequences and assign a qualitative priority of *critical, high, medium, and low* for each loss. Then They calculate threat priority without using weighted numbers or considering likelihood, as it depends on unavailable information, such as software and hardware vendors and versions.

Several papers mention the security mitigation phase. Suleiman and Svetinovic (2013) provide the description of 15 security requirements for AMI (e.g., The system shall prevent unauthorized access to its storage data). Ramis Ferrer et al. (2017) and Khalil et al. (2023) fulfill the security requirements according to ISA/IEC 62443 standard. On the other side, the tool proposed by Martins et al. (2015) uses NIST SP 800-82 Revision 2 (Stouffer et al., 2015b) security controls, and Zahid et al. (2023) use NIST security and privacy controls proposed by Force (2017).

Haider et al. (2019) proposed mitigation techniques for wireless attacks (e.g., Intrusion Detection System (IDS) for man-in-the-middle attacks). The TM tool proposed by Schlegel et al. (2015) is able to propose security controls for the elicited threats (e.g., propose antivirus as mitigation for malware threat). Amro et al. (2023) propose mitigation methods, similar to audit and authorization enforcement, for some MITRE ATT&CK techniques. Al Asif et al. (2021) use defense mechanisms for STRIDE categories, (e.g., Multifactor authentication for device spoofing). On the other side, Fernandez (2016) set security constraints, which we consider a positive action toward threat mitigation. Girdhar et al. (2021) and Liu et al. (2015), both papers include detection solutions, which could be considered as mitigation techniques for the elicited threats. Hacks et al. (2020) mention that mitigation techniques could be added to the modeled system on SecuriCAD, similar to antivirus and firewalls.

Suleiman and Svetinovic (2013) Validates the outcomes of the threat modeling process by including a requirements inspection phase to the SQUARE method, and it mentions that SQUARE provides a checklist-based method to perform such activity, while the details were not discussed in the research. Also, Schlegel et al. (2015) include the TM outcome validation by performing threat modeling manually and using the tool proposed by the authors, then comparing the findings in order to validate the tool outcomes. Khalil et al. (2023) used published tech-

Table 4
ICSs TM methods and techniques.

Reference	TM Method Name	Threat Enumeration Technique	System Mapping Method	Other Used Techniques
Wuyts et al. (2014)	LINDDUN	LINDDUN	DFD	Attack tree
Suleiman et al. (2015)	SG SSTM	SQUARE + SREP		
Suleiman and Svetinovic (2013)	SQUARE			Security template, use case, attack tree
Chen and Huang (2019)	I-SERM	STRIDE+p	DFD	Product Flow Diagram (PFD), Use Case Diagram (UCD), Threat Breakdown Structure + ref. scores (TBS+r), Delphi method, threat tree
Ramis Ferrer et al. (2017)		STRIDE		Qualitative risk model
Ahn et al. (2021)		STRIDE	DFD	CVE, CVSS, attack tree, CAPEC, Matlab
Friedberg et al. (2017)	STPA-SafeSec			
Khan et al. (2017)		STRIDE	DFD	
Fernandez (2016)	Misuse Pattern			Sequence diagram, patterns, activity diagram, class diagram
Haider et al. (2019)		STRIDE	DFD	DREAD
Radoglou-Grammatikis et al. (2019)	CPN-based Threat Model		CPN	Commen Weakness Enumeration (CWE), AlienVault risk assessment model
Zografopoulos et al. (2021a)				Matlab
Stellios et al. (2021)				CVSS, attack path
Girdhar et al. (2021)		STRIDE	DFD	Weighted attack defence tree
Fla et al. (2021)	Microsoft TM Tool	STRIDE	DFD	
Li et al. (2021)		STRIDE	DFD	
Sun et al. (2020)	Pimca Domain Specific Language (DSL)			
Rimsha and Rimsha (2019)	R.I.M.S.H.A. tool			
Schlegel et al. (2015)	System TM tool			
Martins et al. (2015)	An ICSs TM tool			
Vernotte et al. (2018)	SecuriCAD tool			Attack graph
Hacks et al. (2020)	PowerLang DSL			
Zografopoulos et al. (2021b)	OCTAVE Allergo			
Liu et al. (2015)			CPN	
Foldvari et al. (2022)	Impact Analysis Tool			
Kumar et al. (2022)				Attack tree
Amro et al. (2023)	FMECA	ATT&CK		
Lee et al. (2021)	PASTA	STRIDE	DFD	Attack tree
Al Asif et al. (2021)		STRIDE	DFD	
Kim et al. (2022)		STRIDE	DFD	DREAD
Shibly and De Soto (2020)	QTMM	STRIDE	DFD	Attack tree, CVSS
Zahid et al. (2023)		ATT&CK		
Khalil et al. (2023)		STRIDE	DFD	
Valenza et al. (2022)	TAMELESS tool			Attack graph
Jbair et al. (2022)	VTM&CG tool	ATT&CK		Attack tree
Rouland et al. (2021)		STRIDE		Alloy (Formal modeling language)

nical documents to cross-check the outcomes of the threat modeling process.

6.4. ICSs TM methods and techniques

As this research aims to identify the used ICSs TM methods, Table 4 summarizes the names and techniques of the used methods. The table shows that the system mapping method, DFD, is mostly used within STRIDE. Thirteen papers, out the 36 selected paper, use DFD to map the system's information flow as a step in the applications of STRIDE and LINDDUN. On the other side, Ramis Ferrer et al. (2017) propose the use of STRIDE without creating a DFD, instead, the paper uses a component table that provides for each component a description, trust level, and entry points similar to MODBUS, IP, RS232, AND HTTP PORT.

While DFD is broadly used in TM, Colored Petri-Net (CPN) might be a substitute to consider for ICSs. For this reason Table 4 highlights only the DFD and CPN as system mapping methods. The table does not mention the traditional ways of system mapping, similar to the system architecture diagrams. Architecture diagrams are mostly included in TM studies, even when DFD or CPN is applied. These diagrams are fundamental to understanding the network topology and they could be used as a starting point for the information flow mapping. In some threat modeling methods, the architecture diagrams are the only system representation, and no information flow mapping is used.

CPN is an extension of Petri-Net, which can abstractly describe physical phenomena. Petri-Net interprets the relations between the different elements and identifies the information flows. The elements of a Petri-Net are Place, Transition, Connection, and Token. CPN differentiates between the different types of information by providing a different token for each type of information flow (e.g., power flows, data flows, and command flows).

Liu et al. (2015) use CPN to map the information flows of a smart meter, and classify them into two types of information flow: data (users' consumption information and electricity prices from utility companies) and command flows (different commands received from the utility companies and other devices). The provided threat model identifies two types of attacks (attacks on data and attacks on commands). For each of the identified types of attack, they mention targetted information flows, occurring location, and possible attack techniques. The attack techniques were also categorized into two categories (physical attack and cyber attack). Radoglou-Grammatikis et al. (2019) also use CPN and classify the attacks into physical and cyber attacks.

As clarified in Section 2.4, STRIDE is a threat enumeration technique that was initially designed for software TM. For this reason Table 4 does not identify STRIDE as a TM method. Some papers propose combining STRIDE with other techniques, to better represent ICSs. For example, Ahn et al. (2021) complement the TM with attack modeling, combine STRIDE with CVE, CVSS, and attack tree, and use CAPEC to calculate a numerical attack success rate of attacks using MATLAB. The paper

briefly applies the proposed methodology to a power transformer in a digital substation. While such a methodology seems to seek completeness, it leaves the reader with many questions related to the methodology. For example, the research does not provide detailed information about the MATLAB simulation, and the process of choosing CVE, CVSS, and CPEC.

Khan et al. (2017) mention that there is no standard methodology defined in the literature for applying STRIDE, and propose a methodology for its application on CPSs. Identifying a list of possible threat consequences is proposed by Khan et al. (2017) as a first step in threat analysis. Then, the study uses STRIDE-per-element followed by STRIDE-per-interaction, to complement the list of elicited threats. Kim et al. (2022) is another example of studies that combine STRIDE-per-element and STRIDE-per-interaction a in single methodology.

Li et al. (2021) is another example of studies that use STRIDE and apply it to the workflow of operators and control engineers who setup the webserver of TIA portal.[25] Li et al. (2021) use STRIDE per element, without explicitly mentioning the term, however, it did not use the famous STRIDE threat categories for mapping the DFD elements to threats. The paper also added a new STRIDE analysis element named "task scenario", which includes authentication, network and access configuration, information gathering by the operator, client authentication, and operator remote work tasks. Li et al. (2021) map the task scenarios to "usability factors" that are selected from previous literature based on CPSs needs. Examples of chosen usability factors are efficiency, safety, availability, and accuracy. The paper then categorizes the selected usability factors into 3 categories, environment-related, application-related, and user-related.

Another example of STRIDE application is presented by Haider et al. (2019), where the authors combine STRIDE with DREAD for the analysis and prioritization of wireless attacks on AMI. The study identifies 5 related threats (DOS, DDOS, FDI, De-pseudonymization, and MITM), and applies STRIDE to those threats. STRIDE is also used by Girdhar et al. (2021), in combination with a weighted attack defense tree that describes the attack objectives of the adversary.

Lee et al. (2021), added by the third iteration of this SLR, combine STRIDE for threat enumeration within the *Process of Attack Simulation and Threat Analysis* (PASTA) TM. The research justifies its choice for PASTA, as it can incorporate attacker-centric and data-centric to its asset-centric approach. PASTA applies seven threat modeling stages: define objectives, define technical scope, system/application decomposition, threat analysis, vulnerabilities and weakness analysis, attack modeling, and risk and impact analysis. The study uses an attack tree and STRIDE to analyze the data flows.

Shibly and De Soto (2020) propose a framework for Quantitative Threat Modeling Methods (QTMM) that is composed of 6 steps: define use case/problem statement, define DFD, map sTRIDE elements into DFD, identify threats, perform Quantitative risk assessment and plan risk mitigation strategies. The research combines sTRIDE with the attack tree and uses the CVSS score and risk propagation technique to calculate the risk. The paper proposes also adding risk mitigation into the attack tree.

As the STRIDE application can be automated using Microsoft Threat Modeling Tool, Fla et al. (2021) propose a smart grid template for the tool. The study created a list of threats based on the literature, existing templates, and cyber-attacks on OT. The tool template is configured to use stride-per-interaction. While the research also criticized STRIDE for not including a method for threat prioritization. STRIDE was also used with the context of formal method in Rouland et al. (2021), where it was used to elicit threats using technology-independent specification language.

[25] Siemens TIA portal is used by engineers and operators to configure PLCs during and after system deployment.

The study Chen and Huang (2019) uses a threat enumeration method based on STRIDE with privacy aspect enhancements (i.e., named STRIDE+p), as a part of a new TM method named I-SERM. As per Chen and Huang (2019), STRIDE+p combines the features of STRIDE and LINDDUN, however, the methodology is not detailed enough to ease the use of the proposed method. The I-SERM method considers risk management to prioritize threats and proposes a process of seven steps to apply the model.

The use of STRIDE was justified by Girdhar et al. (2021), as STRIDE is considered the most mature threat modeling approach. The paper mentions that STRIDE evaluates the detailed system design by building DFD and identifies cyber threats against each system entity. Moreover, STRIDE is comprehensive and analyzes security properties against each system component, while ensuring system security at the component level. Also, as per Girdhar et al. (2021), STRIDE is widely adopted as a cyber security framework to secure critical infrastructure from cyber attacks in CPSs.

It is interesting to mention that some papers (e.g., Khan et al. (2017)), map analog physical components on the DFD as an external entity, while there might be a need to differentiate between an external entity and a physical component that does not have a digital connection to the system. Also, differentiating between data flows, command flows, and power flows might be important information to map on the diagrams of an ICS. As some flows might be more critical than others (e.g., control flows), which was acheived by CPN in the studies of Radoglou-Grammatikis et al. (2019) and Liu et al. (2015). On the other hand, Khalil et al. (2023) used a colored DFD, to differentiate between three types of processes: electrical power, measurement, and software-related. The paper also identifies physical processes, and analog connections in the DFD, to make it more comprehensive for the involved stakeholders.

We have also noticed that the third search iteration included research that used MITRE ATT&Ck for threat enumeration (Zahid et al. (2023), Amro et al. (2023) and Jbair et al. (2022)). Using a specific ICSs attacks database similar to MITRE ATT&CK, which is constantly updated, could enrich the threat elicitation process. The VueOne Threat Modeller and Code Generator Tool (VTM&CG) proposed by Jbair et al. (2022) also incorporate the use of attack trees.

Some papers propose different TM methods other than STRIDE. For instance, Suleiman and Svetinovic (2013) evaluate the Security Quality Requirements Engineering (SQUARE) method, which consists of nine steps: agree on definitions, identify security goals, develop artifacts to support security, requirements definitions, perform risk assessment, select requirements elicitation technique, elicit the security requirements, categorize the security requirements, prioritize the security requirements, and inspect the security requirements. While most of the TM methods propose threat prioritization, SQUARE prioritizes the security requirements instead of the threats, which require extra effort. Suleiman et al. (2015) introduce a method named Smart Grid Systems Security Threat Model (SG SSTM), which proposes a reference architecture of a smart grid and categorizes its security threats into 6 categories: network availability, data integrity, data confidentiality and information privacy, smart grid key operation, inter-organizational collaborations, and relationships, and the resilience and safety of SG infrastructure. While the paper mentions that the threats were elicited using 2 threat analysis methods: SQUARE (Suleiman and Svetinovic, 2013) and Security Requirements Engineering Process (SREP) (Wang et al., 2009) methods, it just share the outcomes of the threat elicitation process without details on the elicitation process.

TM methods mostly use a bottom-up approach, where threats are used to derive the security requirements. However, methods with a safety and security focus, similar to STPA-SAFESec, prefer to use the top-down approach. A top-down approach identifies the unacceptable system losses, and the analysis moves from general to specific (Young and Leveson, 2014). For instance, STPA-SAFESec starts with defining the control layer where the system losses, hazards, and control loops

are identified. The method also includes refining safety and security constraints and defining hazard scenarios that help in identifying mitigation strategies.

Vernotte et al. (2018) provide a load-balancing centered smart grid reference architecture model, and perform a TM using SecuriCAD. The study defines the tool as *"a modeling framework and calculation engine that estimates the cyber security of systems-of-systems-level architectures"*. Using SecuriCAD, the paper models the system architecture, including data flows, and identifies its security characteristics (e.g., host firewall enabled). The tool then creates attack graphs for the modeled system and calculates the shortest paths and Time-To-Compromise for attacks.

Rimsha and Rimsha (2019) briefly describes the «Risk Identification and Management Security Host-based Appliance» (R.I.M.S.H.A.) tool. The tool performs TM and risk assessment based on a quantitative risk assessment method. Schlegel et al. (2015) propose a TM tool based on a data model of component, threat, impact, and security control. Martins et al. (2015) propose a tool that can identify potential threats of a CPS based on MetaGME modeling language. Foldvari et al. (2022) propose an impact analysis tool that uses the error propagation analysis while considering both CPSs safety and security. The tool explores the impact of a cyber attack on CPSs and highlights critical control paths, which help in choosing/validating countermeasures.

Some papers propose domain-specific modeling languages. PowerLang (Hacks et al., 2020) is a MAL-based domain-specific language for modeling IT and OT infrastructures in the power domain. Hacks et al. (2020) reuse two existing MAL-based DSL (corelang, sclLang), and propose (icsLang) to fill the gaps in the reused languages. Pimca, an attack-centric domain-specific threat modeling language, is proposed by Sun et al. (2020). The Pimca DSL was originally developed by the Directorate General of Armaments (DGA), French Ministry of Armed Forces.

6.5. Other analyzed aspects

As a part of this qualitative analysis, the recommended development life cycle stage for TM was analyzed in each of the selected papers. Wuyts et al. (2014) mention that "*Threat modeling can be applied at different levels of abstraction, depending on the granularity of assets considered*". As per Wuyts et al. (2014), the assets are more abstract at the requirement phase, while they become more tangible during the architectural design. While some papers (e.g., Ramis Ferrer et al. (2017); Ahn et al. (2021); Martins et al. (2015); Girdhar et al. (2021)) explicitly mention the design phase for TM application, other papers (e.g., Suleiman et al. (2015); Suleiman and Svetinovic (2013); Friedberg et al. (2017); Khan et al. (2017)) just mention that TM is applied at the early stages of the system development life cycle. The study presented by Schlegel et al. (2015) highlights the possibility of running TM at the operation phase for threat mitigation, while it mentions that running TM at the design and implementation phases is more cost-effective.

Another aspect that this SLR looked into is the required expert knowledge for each of the selected methods. Some methods explicitly mention the need for expert knowledge to run them. For example, Suleiman and Svetinovic (2013) mention that the application of the SQUARE method requires some level of domain expertise in order to identify potential threats, and propose that such action be done through a workshop. Experts are also required to assess the threats' impacts and their criticality. For example, the Delphi method, which is based on expert knowledge, is proposed for threat prioritization by Chen and Huang (2019) based on the impact and likelihood of the threats. Assessing the impacts of the threats by experts is also proposed by Ramis Ferrer et al. (2017) and Zografopoulos et al. (2021b). Khan et al. (2017) identify the threat consequences by domain experts. System losses and hazards are also identified by experts in the case of Friedberg et al. (2017). The proposed tool by Schlegel et al. (2015) targets security professionals, as sufficient security knowledge is a prerequisite. On the other hand, Vernotte et al. (2018) mention that the securiCAD tool has collected quantitative data from different sources, so there is no security expertise requirement from the tool user.

7. Research gaps and future directions

This SLR discussed the different aspects of ICSs threat modeling. The analysis shows that the ICSs TM inherits the characteristics of software TM as a general domain. The ICSs TM is currently more focused on the threat elicitation stage and does not provide clear procedures for other stages, while a clear description for all TM stages is a vital requirement for practicing threat modeling in real settings. Also, the selected threat modeling papers do not discuss the technical and organizational challenges and how to overcome them regarding the application of these methods. These notes go with the finding of Yskout et al. (2020), which also concluded that threat modeling, as an engineering discipline, is at a low level of maturity even in the software security domain where TM originated. The SLR findings are in line with other findings of Yskout et al. (2020), as this SLR identifies that there is a lack of published industrial case studies and experience reports related to ICSs TM.

TM is an interactive activity built on communication between various teams with different perspectives and goals, such as system developers, system administrators, security experts, and domain professionals. Compared to software-based systems, the development and maintenance of CPSs require more diverse expertise, ranging from experts in physical processes to system developers. This communication aspect has not been studied in the selected papers. It is unclear how to effectively construct such communication to ensure that different teams have a common understanding of the discussed threats and their implications. Interdisciplinary studies that address the intersection of technology and human perspectives are highly required in this problem domain. As a more specific suggestion, the use of gamification techniques might be considered as a potential instrument for team communication in TM processes, which could be studied in future research.

In this research, we noticed that "threat modeling" is also used for cases in which a systematic method is not followed. For example, the study of Soltan et al. (2018) has a threat modeling section, while it does not follow a systematic threat enumeration approach. It could be considered as a listing of some possible attacks based on a defined scenario, however, it just focuses on a specific type of attack without enumerating all possible attacks. Therefore, we excluded such studies from this SLR (see Section 4.2.2). We contemplate that a threat modeling method should be applicable to a wider group of systems and cover various attack types rather than being very specific to an attack type or target system.

This SLR identified two types of validation for ICSs TM. First, validating the used ICSs TM method which aims the evaluate aspects similar to the effectiveness and correctness of the method. Such validation could be done through qualitative or quantitative research methods. The second type of validation is validating the outcomes of each ICSs TM run regardless of the used methodology. The TM outcomes validation could be considered as the "quality assurance" for the TM process. It aims to ensure that the critical threats were elicited and suitable security requirements were proposed. This second type of validation might be done through a peer review or it might be automated. The provided analysis shows that there is a clear gap in both validation types, which also goes with the findings of Tuma et al. (2018).

While STRIDE seems to be the most mature threat modeling method, this SLR shows that there is no clear understanding of how to document the threats elicited using STRIDE. While many of the selected studies used STRIDE, they do not use a common language to describe the threat elicited by STRIDE. For example, Chen and Huang (2019) elicit STRIDE threats through a table, where the threat elicitation phase was merged with the risk assessment, and no detailed description is provided for each threat. On the other hand, Ramis Ferrer et al. (2017) elicit the threats with a description based on a chosen CVE related to each STRIDE threat. Also, the level of expertise for identifying threat

consequences on ICSs seems to be a limitation toward the automation of ICSs TM, as identifying the criticality of an ICSs threat should be related to the impact the threat might cause on the system. This raises the question of the effectiveness of using threat libraries and knowledge bases similar to MITRE ATTA&CK.

ICSs play a key role in most of the critical infrastructures that require special consideration. The design of such systems should consider ensuring safety and security. If the target system processes information about individuals, data privacy would be another concern as a system objective. However, this study shows a need for creating mature and easy-to-adapt ICSs TM methods that cover all those aspects simultaneously.

Security threat models help derive preventive and detective controls, such as authentication, access control, and network controls. Effective reactive controls, such as incident handling, depend on the forensic readiness of the system, which requires considering forensic artifact extraction and analysis requirements early in development. Experts can handle these requirements using current models (e.g., non-repudiation in STRIDE). However, the current models do not provide a systematic instrument to address forensics readiness requirements. TM techniques can be extended to cover this important aspect of critical infrastructure, as described by Iqbal et al. (2020).

All the selected papers use TM during the design phase of an ICS. Although this finding aligns with the common perception of TM as an early-stage development instrument, we contemplate that the knowledge created in TM can be utilized in the later steps of SDLC. For instance, while conducting penetration testing at the verification stage. As cyber-physical systems have various types of interfaces, protocols, and data flows, it is very difficult to define a scope and proper course of action for penetration testing endeavors. Security experts do not perform this task systematically despite the fact that enough human resources are not available in most cases. TM can act as an instrument for identifying and prioritizing the system components/information flows that would be addressed during the penetration testing within the given resource limitations. Additionally, the utilization of TM enables knowledge transfer between the early and late-stage steps.

Threat modeling methods reviewed in this paper usually perceive the target system as static. Although some of them elaborate on the malicious actors while eliciting the threats, they do not consider the interaction between those actors and defenders. It is assumed that the defenders do not apply more proactive countermeasures such as moving target defence (i.e., changing the attack surface to misguide the attackers). It is a promising research path to incorporate game theory into threat modeling to reconcile the attack and defence actions and consider dynamic defensive actions (e.g., moving target defence) in such modeling formulation (Tan et al., 2022, 2023).

It is obvious that we may expect more ICSs components that benefit from artificial intelligence (AI) as various research studies demonstrate the applicability of AI in addressing the problem domains where cyber-physical systems are prevalent (Raza and Khosravi, 2015; Stetco et al., 2019). A huge body of research proposed adversarial attacks that threaten the AI-based models (Papernot et al., 2016). A very limited number of studies have focused on systematically identifying such attacks using threat models (Bitton et al., 2022). More importantly, providing a holistic threat model that synthesizes data-centric adversarial attacks and conventional cyber attacks and applying it to ICSs constitutes an interesting research direction.

Threat modeling is a domain in which advancement is driven by both academia and industry. Although Microsoft's SDLC practices introduced this notion to the software development community, academia has systematized it for ICSs (Khan et al., 2017; Khalil et al., 2023) and adapted it to privacy threat modeling (Wuyts et al., 2014). More cooperation between both parties is needed to demonstrate the applicability of the methods in various application areas. We admit that the proper validation of the threat modeling results is very hard to achieve as the nature of the problem does not enable to maintain ground truth data. However, still, threat libraries and knowledge bases can be enhanced cooperatively. Academia can put more effort into developing qualitative or quantitative validation methods.

8. Threats to validity

This SLR highlighted multiple research gaps in the ICSs TM domain. However, the results of this study are not free from validity threats that might impact the main findings.

The inclusion-exclusion criteria limited the number of studies included in this SLR. Setting the scope of the study to the ICSs application area excluded valuable CPSs TM methods. For example, de Souza et al. (2020) propose a safety and security-related threat model by extending STPA with STRIDE, but it was excluded from our literature review because it used an E-Voting system as an application area. Some papers were excluded from the list while they were related to ICSs, as they did not meet our criteria for ICSs. For example, Hammad et al. (2017) propose a Vulnerable-Link Adaptive Avoidance (VLAA) algorithm that uses a formal vulnerability metric for communication links in CPSs. The paper was excluded from our list, while its application domain is related to Software Defined Networks (SDN) in power systems, as it is related to communication and network applications and does not use system components related to how we identify an ICS. Also, most of the formal TM-related papers were excluded, as they tend to analyze one or two types of attacks, which did not meet our selection criteria. These facts prevented having a more exhaustive list of TM methods in our SLR.

We also noticed that this SLR list did not include some well-known threat modeling methods (e.g., TRIKE, OCTAVE), and the included OCTAVE-related paper, Zografopoulos et al. (2021b), was found through snowballing. Such missing results might be related to the keywords used in the search criteria, as these methods are mostly categorized as *risk assessment* methods, and sometimes they are also called *security assessment* methods. It was also noticed that in safety-related papers, the term *security analysis* is used instead of threat modeling (e.g., Friedberg et al. (2017)). Also, we found that TM is sometimes named *whiteboard hacking* (Yskout et al., 2020). Such findings might indicate a need for adjusting the search criteria in similar future research to include the keywords: risk assessment, security analysis, security assessment, and whiteboard hacking. However, the application of snowballing reduced the impact of limited keywords.

9. Conclusion

This research identified various ICSs threat modeling studies, suggesting that each method has its focus and covered aspects. The decision to choose a relevant ICSs TM method is based on the practitioners' needs. There are some considerably mature methods (i.e., STRIDE and LINDDUN), while most of the proposed ICSs methods still need further validation and testing. Although many methods address safety, security, and privacy aspects separately, it is time to propose a simple, well-documented, and validated framework that covers all these aspects for ICSs.

CRediT authorship contribution statement

Shaymaa Mamdouh Khalil: Conceptualization, Data curation, Investigation, Methodology, Visualization, Writing – original draft, Writing – review & editing. **Hayretdin Bahsi:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Tarmo Korõtko:** Funding acquisition.

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.

Data availability

Data will be made available on request.

Declaration of generative AI and AI-assisted technologies in the writing process

During the final revision of this work, the author(s) used Bard (Google chat-based AI) in order to review the grammar and language of some sentences. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Acknowledgements

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

Adam Shostack, 2013. Threat Modeling Designing for Security, vol. 53. John Wiley & Sons. Incorporated. https://ebookcentral.proquest.com/lib/tuee/reader.action?docID=1629177. arXiv:1011.1669v3.

Ahn, B., Kim, T., Smith, S.C., Youn, Y.W., Ryu, M.H., 2021. Security Threat Modeling for Power Transformers in Cyber-Physical Environments, pp. 1–5.

Al Asif, M.R., Hasan, K.F., Islam, M.Z., Khondoker, R., 2021. Stride-based cyber security threat modeling for IoT-enabled precision agriculture systems. In: 2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI). IEEE, pp. 1–6.

Alberts, C., Dorofee, A., Stevens, J., Woody, C., 2003. Introduction to the OCTAVE Approach. Technical Report Carnegie-Mellon Univ. Pittsburgh PA Software Engineering Inst.

Alexander, O., Belisle, M., Steele, J., 2020. MITRE ATT&CK for industrial control systems: design and philosophy. https://collaborate.mitre.org/attackics/img_auth.php/3/37/ATT%26CK_for_ICS_-_Philosophy_Paper.pdf.

Allodi, L., Etalle, S., 2017. Towards realistic threat modeling: attack commodification, irrelevant vulnerabilities, and unrealistic assumptions. In: SafeConfig 2017 - Proceedings of the 2017 Workshop on Automated Decision Making for Active Cyber Defense, Co-Located with CCS 2017, pp. 23–26. https://doi.org/10.1145/3140368.3140372. arXiv:1801.04569.

Amro, A., Gkioulos, V., Katsikas, S., 2023. Assessing cyber risk in cyber-physical systems using the ATT&CK framework. ACM Trans. Priv. Secur. 26. https://doi.org/10.1145/3571733.

Bernsmed, K., Jaatun, M.G., 2019. Threat modelling and agile software development: identified practice in four Norwegian organisations. In: 2019 International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2019. https://doi.org/10.1109/CyberSecPODS.2019.8885144.

Bitton, R., Avraham, D., Klevansky, E., Mimran, D., Brodt, O., Lehmann, H., Elovici, Y., Shabtai, A., 2022. Adversarial machine learning threat analysis in open radio access networks. arXiv preprint. arXiv:2201.06093.

Burmester, M., Magkos, E., Chrissikopoulos, V., 2012. Modeling security in cyber–physical systems. Int. J. Crit. Infrastructures Prot. 5, 118–126. https://doi.org/10.1016/j.ijcip.2012.08.002.

Caltagirone, S., Pendergast, A., Betz, C., 2013. The diamond model of intrusion analysis. Technical Report. Center for Cyber Intelligence Analysis and Threat Research Hanover Md.

Chen, Y.T., Huang, C.C., 2019. Determining information security threats for an IoT-based energy Internet by adopting software engineering and risk management approaches. Inventions, 4. https://doi.org/10.3390/inventions4030053.

Cherepanov, A., 2017. WIN32/INDUSTROYER a New Threat for Industrial Control Systems.

Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W., 2011. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requir. Eng. 16, 3–32. https://doi.org/10.1007/s00766-010-0115-7.

Fernandez, E.B., 2016. Threat modeling in cyber-physical systems. In: Proceedings - 2016 IEEE 14th International Conference on Dependable, Autonomic and Secure Computing, DASC 2016, 2016 IEEE 14th International Conference on Pervasive Intelligence and Computing, PICom 2016, 2016 IEEE 2nd International Conference on Big Data, pp. 448–453. https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2016.89.

Fla, L.H., Borgaonkar, R., Tondel, I.A., Gilje Jaatun, M., 2021. Tool-assisted threat modeling for smart grid cyber security. In: 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment. CyberSA 2021. https://doi.org/10.1109/CyberSA52016.2021.9478258.

Foldvari, A., Biczok, G., Kocsis, I., Gonczy, L., Pataricza, A., 2022. Impact Assessment of IT Security Breaches in Cyber-Physical Systems: Short paper, 1–4.

Force, J.T., 2012. Guide for Conducting Risk Assessments.

Force, J.T., 2017. Security and Privacy Controls for Information Systems and Organizations. NIST Special Publication.

Friedberg, I., McLaughlin, K., Smith, P., Laverty, D., Sezer, S., 2017. STPA-SafeSec: safety and security analysis for cyber-physical systems. J. Inf. Secur. Appl. 34, 183–196. https://doi.org/10.1016/j.jisa.2016.05.008.

Girdhar, M., Hong, J., Lee, H., Song, T.-J., 2021. Hidden Markov models based anomaly correlations for the cyber-physical security of EV charging stations. IEEE Trans. Smart Grid 13 (5), 3903–3914. https://doi.org/10.1109/tsg.2021.3122106.

Goldsmith, J., Levinson, D., Harvard, S., Review, L., May, N., 1890. The Harvard law review association. Harvard Law Rev. 4, 193–220.

Hacks, S., Katsikeas, S., Ling, E., Lagerström, R., Ekstedt, M., 2020. powerLang: a probabilistic attack simulation language for the power domain. Energy Inform. 3. https://doi.org/10.1186/s42162-020-00134-4.

Haider, M.H., Saleem, S.B., Rafaqat, J., Sabahat, N., 2019. Threat modeling of wireless attacks on advanced metering infrastructure. In: MACS 2019 - 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics, Proceedings. https://doi.org/10.1109/MACS48846.2019.9024779.

Hajrić, A., Smaka, T., Baraković, S., Husić, J.B., 2020. Methods, methodologies, and tools for threat modeling with case study. Telfor J. 12. https://doi.org/10.5937/telfor2001056H.

Hammad, E., Farraj, A., Kundu, D., 2017. Communication Links Vulnerability Model for Cyber Security Mitigation. LNICST, vol. 184.

Howard, M., LeBlanc, D., 2002. Writing Secure Code. Microsoft Press.

Hutchins, E.M., Cloppert, M.J., Amin, R.M., et al., 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Lead. Issues Inform. Warf. Secur. Res. 1, 80.

Iqbal, A., Olegard, J., Ghimire, R., 2020. Digital forensic evidence-the missing link in threat modeling. In: 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy. ICDABI 2020. https://doi.org/10.1109/ICDABI51230.2020.9325650.

Jamil, A.M., Ben Othmane, L., Valani, A., 2022. Threat Modeling of Cyber-Physical Systems in Practice. Technical Report. https://otter.ai/. arXiv:2103.04226v1.

Jbair, M., Ahmad, B., Maple, C., Harrison, R., 2022. Threat modelling for industrial cyber physical systems in the era of smart manufacturing. Comput. Ind. 137, 103611.

Johnson, P., Lagerström, R., Ekstedt, M., 2018. A Meta Language for Threat Modeling and Attack Simulations, p. 8.

Khalil, S.M., Bahsi, H., Ochieng'Dola, H., Korõtko, T., McLaughlin, K., Kotkas, V., 2023. Threat modeling of cyber-physical systems-a case study of a microgrid system. Comput. Secur. 124, 102950.

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

Kim, K.H., Kim, K., Kim, H.K., 2022. Stride-based threat modeling and dread evaluation for the distributed control system in the oil refinery. ETRI J. 44, 991–1003.

Kitchenham, B., Charters, S., et al., 2007. Guidelines for Performing Systematic Literature Reviews in Software Engineering.

Kriaa, S., Pietre-Cambacedes, L., Bouissou, M., Halgand, Y., 2015. A Survey of Approaches Combining Safety and Security for Industrial Control Systems.

Kumar, R., Kela, R., Singh, S., Trujillo-Rasua, R., 2022. APT attacks on industrial control systems: a tale of three incidents. Int. J. Crit. Infrastructures Prot. 37, 100521. https://doi.org/10.1016/j.ijcip.2022.100521.

Lallie, H.S., Debattista, K., Bal, J., 2020. A review of attack graph and attack tree visual syntax in cyber security. Comput. Sci. Rev. 35, 100219.

Lee, C.C., Tan, T.G., Sharma, V., Zhou, J., 2021. Quantum computing threat modelling on a generic cps setup. In: Applied Cryptography and Network Security Workshops: ACNS 2021 Satellite Workshops, Proceedings. AIBlock, AIHWS, AIoTS, CIMSS, Cloud S&P, SCI, SecMT, and SiMLA, Kamakura, Japan, June 21–24, 2021. Springer, pp. 171–190.

Li, K., Rashid, A., Roudaut, A., 2021. Vision: security-usability threat modeling for industrial control systems. In: ACM International Conference Proceeding Series, pp. 83–88. https://doi.org/10.1145/3481357.3481527.

Ling, E., Lagerström, R., Ekstedt, M., 2020. A systematic literature review of information sources for threat modeling in the power systems domain. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer Science and Business Media Deutschland GmbH, pp. 47–58. https://doi.org/10.1007/978-3-030-58295-1_4.

Liu, X., Zhu, Peidong Z.Y., Chen, K., 2015. A collaborative intrusion detection mechanism against false data injection attack in advanced metering infrastructure. Smart Grid Technol. 6, 2435–2443. https://doi.org/10.1017/9781108566506.011.

Luo, F., Jiang, Y., Zhang, Z., Ren, Y., Hou, S., 2021. Threat analysis and risk assessment for connected vehicles: a survey. Secur. Commun. Netw. 2021, 1–19. https://doi.org/10.1155/2021/1263820.

Maheshwari, V., Prasanna, M., 2016. Integrating risk assessment and threat modeling within SDLC process. In: 2016 International Conference on Inventive Computation Technologies, vol. 1. ICICT, pp. 1–5.

Mai, P.X., Goknil, A., Shar, L.K., Pastore, F., Briand, L.C., Shaame, S., 2018. Modeling security and privacy requirements: a use case-driven approach. Inf. Softw. Technol. 100, 165–182. https://doi.org/10.1016/j.infsof.2018.04.007.

Marksteiner, S., Ramler, R., Sochor, H., 2019. Integrating threat modeling and automated test case generation into industrialized software security testing. In: Pervasive-Health: Pervasive Computing Technologies for Healthcare. https://doi.org/10.1145/3360664.3362698. arXiv:1911.06594.

Martins, G., Bhatia, S., Koutsoukos, X., Stouffer, K., Tang, C., Candell, R., 2015. Towards a systematic threat modeling approach for cyber-physical systems. In: Proceedings - 2015 Resilience Week. RSW 2015, pp. 114–119. https://doi.org/10.1109/RWEEK.2015.7287428.

Mead, N.R., Stehney, T., 2005. Security Quality Requirements Engineering (SQUARE) Methodology. ACM SIGSOFT Software Engineering Notes, vol. 30.

Morana, M.M., UcedaVelez, T., 2015. Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis. John Wiley & Sons. https://learning.oreilly.com/library/view/risk-centric-threat/9780470500965/c03.xhtml#c03_level1_1.

Nweke, L.O., Wolthusen, S.D., 2020. A review of asset-centric threat modelling approaches. Int. J. Adv. Comput. Sci. Appl., 1–6. https://doi.org/10.14569/ijacsa.2020.0110201.

Papernot, N., McDaniel, P., Sinha, A., Wellman, M., 2016. Towards the science of security and privacy in machine learning. arXiv preprint. arXiv:1611.03814.

Paverd, A., Martin, A., Brown, I., 2014. Smart grid security: second international workshop. In: SmartGridSec 2014, Revised Selected Papers. Munich, Germany, February 26, 2014. https://doi.org/10.1007/978-3-319-10329-7_1.

Radoglou-Grammatikis, P., Sarigiannidis, P., Giannoulakis, I., Kafetzakis, E., Panaousis, E., 2019. Attacking IEC-60870-5-104 SCADA systems. In: Proceedings - 2019 IEEE World Congress on Services, SERVICES 2019, pp. 41–46. https://doi.org/10.1109/SERVICES.2019.00022.

Rak, M., Salzillo, G., Romeo, C., 2020. Systematic IoT penetration testing: ALEXA case study. In: CEUR Workshop Proceedings, vol. 2597, pp. 190–200.

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. In: Proceedings IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society 2017-Janua, pp. 3498–3505. https://doi.org/10.1109/IECON.2017.8216592.

Raza, M.Q., Khosravi, A., 2015. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. Renew. Sustain. Energy Rev. 50, 1352–1372.

Research, E., 2022. Industroyer2: industroyer reloaded | WeLiveSecurity. https://www.welivesecurity.com/2022/04/12/industroyer2-industroyer-reloaded/.

Rimsha, A., Rimsha, K., 2019. Development of threat modeling and risk management tool in automated process control system for gas producing enterprise. In: Proceedings - 2019 21st International Conference "Complex Systems: Control and Modeling Problems". CSCMP 2019 2019-Septe, pp. 596–599.

Rouland, Q., Hamid, B., Jaskolka, J., 2021. Specification, detection, and treatment of stride threats for software components: modeling, formal methods, and tool support. J. Syst. Archit. 117, 102073.

Saitta, P., Larcom, B., Eddington, M., 2005. Trike v.1 methodology document [draft]. http://dymaxion.org/trike/Trike_v1_Methodology_Documentdraft.pdf.

Salzillo, G., Rak, M., Moretta, F., 2020. Threat modeling based penetration testing: the open energy monitor case study. In: ACM International Conference Proceeding Series. https://doi.org/10.1145/3433174.3433181.

Scandariato, R., Wuyts, K., Joosen, W., 2015. A descriptive study of Microsoft's threat modeling technique. Requir. Eng. 20, 163–180. https://doi.org/10.1007/s00766-013-0195-2.

Schlegel, R., Obermeier, S., Schneider, J., 2015. Structured system threat modeling and mitigation analysis for industrial automation systems. In: Proceeding - 2015 IEEE International Conference on Industrial Informatics. INDIN 2015, pp. 197–203. https://doi.org/10.1109/INDIN.2015.7281734.

Sequeiros, J.B., Chimuco, F.T., Samaila, M.G., Freire, M.M., Inácio, P.R., 2020. Attack and system modeling applied to IoT, cloud, and mobile ecosystems: embedding security by design. ACM Comput. Surv. 53, 1–32.

Shevchenko, N., Chick, T.A., Riordan, P.O., Scanlon, T.P., Woody, C., 2018a. Threat Modeling: a Summary of Available Methods.

Shevchenko, N., Frye, B., Woody, C., 2018b. Threat modeling for cyber-physical system-of-systems: methods evaluation. https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_526372.pdf.

Shibly, M., De Soto, B.G., 2020. Threat modeling in construction: an example of a 3d concrete printing system. In: ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction. IAARC Publications, pp. 625–632.

Shostack, A., 2008. Experiences Threat Modeling at Microsoft. CEUR Workshop Proceedings, vol. 413, pp. 1–11.

Sion, L., Yskout, K., Van Landuyt, D., Joosen, W., 2018. Risk-based design security analysis. In: Proceedings of the 1st International Workshop on Security Awareness from Design to Deployment, pp. 11–18.

Soares Cruzes, D., Gilje Jaatun, M., Bernsmed, K., Tondel, I.A., 2018. Challenges and experiences with applying Microsoft threat modeling in agile development projects. In: Proceedings - 25th Australasian Software Engineering Conference. ASWEC 2018, pp. 111–120. https://doi.org/10.1109/ASWEC.2018.00023.

Soltan, S., Mittal, P., Poor, H.V., 2018. BlackIoT: IoT botnet of high wattage devices can disrupt the power grid. In: Proceedings of the 27th USENIX Security Symposium, pp. 15–32.

Souppaya, M., Scarfone, K., 2016. Guide to data-centric system threat modeling. Technical Report. National Institute of Standards and Technology.

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. J. Inf. Secur. Appl. 55. https://doi.org/10.1016/j.jisa.2020.102620.

Stellios, I., Kotzanikolaou, P., Grigoriadis, C., 2021. Assessing IoT enabled cyber-physical attack paths against critical systems. Comput. Secur. 107, 102316. https://doi.org/10.1016/j.cose.2021.102316.

Stetco, A., Dinmohammadi, F., Zhao, X., Robu, V., Flynn, D., Barnes, M., Keane, J., Nenadic, G., 2019. Machine learning methods for wind turbine condition monitoring: a review. Renew. Energy 133, 620–635.

Steven, J., 2010. Threat modeling-perhaps it's time. IEEE Secur. Priv. 8, 83–86.

Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M., Hahn, A., 2015a. Guide to industrial control systems (ICS) security NIST special publication 800-82 revision 2. NIST special publication 800-82 rev 2, 1–157. http://industryconsulting.org/pdfFiles/NIST_Draft-SP800-82.pdf.

Stouffer, K., Pillitteri, V., Lightman, S., Abrams, M., Hahn, A., 2015b. NIST Special Publication 800-82 Revision 2 Guide to Industrial Control Systems (ICS) Security. https://doi.org/10.6028/NIST.SP.800-82r2.

Suleiman, H., Alqassem, I., Diabat, A., Arnautovic, E., Svetinovic, D., 2015. Integrated smart grid systems security threat model. Inf. Sci. 53, 147–160. https://doi.org/10.1016/j.is.2014.12.002.

Suleiman, H., Svetinovic, D., 2013. Evaluating the effectiveness of the security quality requirements engineering (SQUARE) method: a case study using smart grid advanced metering infrastructure. Requir. Eng. 18, 251–279. https://doi.org/10.1007/s00766-012-0153-4.

Sun, T., Drouot, B., Golra, F., Champeau, J., Guerin, S., Roux, L.L., Mazo, R., Teodorov, C., Aertryck, L., Hostis, B., Sun, T., Drouot, B., Golra, F., Champeau, J., Guerin, S., Domain, A., Sun, T.N., Drouot, B., Golra, F.R., 2020. A Domain-Specific Modeling Framework for Attack Surface Modeling.

Suo, D., Siegel, J.E., Sarma, S.E., 2018. Merging safety and cybersecurity analysis in product design. IET Intell. Transp. Syst. 12, 1103–1109.

Süren, E., Heiding, Fredrik, Olegård, J., Lagerström, R., 2023. PatrIoT: practical and agile threat research for IoT. Int. J. Inf. Secur. 22, 213–233. https://doi.org/10.1007/s10207-022-00633-3.

Tan, J., Jin, H., Hu, H., Hu, R., Zhang, H., Zhang, H., 2022. WF-MTD: evolutionary decision method for moving target defense based on Wright-Fisher process. IEEE Trans. Dependable Secure Comput., 1–14.

Tan, J., Jin, H., Zhang, H., Zhang, Y., Chang, D., Liu, X., Zhang, H., 2023. A survey: when moving target defense meets game theory. Comput. Sci. Rev. 48, 100544.

Tarandach, I., Coles, J.M., 2020. Threat Modeling: A Practical Guide for Development Teams.

Tuma, K., Calikli, G., Scandariato, R., 2018. Threat analysis of software systems: a systematic literature review. J. Syst. Softw. 144, 275–294. https://doi.org/10.1016/j.jss.2018.06.073.

Tuma, K., Scandariato, R., Widman, M., Sandberg, C., 2017. Towards security threats that matter. In: Computer Security. Springer, pp. 47–62.

UcedaVelez, T., Morana, M.M., 2015. Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis. John Wiley & Sons.

Uzunov, A.V., Fernandez, E.B., 2014. An extensible pattern-based library and taxonomy of security threats for distributed systems. Comput. Stand. Interfaces 36, 734–747. https://doi.org/10.1016/j.csi.2013.12.008.

Valenza, F., Karafili, E., Steiner, R.V., Lupu, E.C., 2022. A hybrid threat model for smart systems. IEEE Trans. Dependable Secure Comput. 20 (5), 4403–4417.

Vernotte, A., Välja, M., Korman, M., Björkman, G., Ekstedt, M., Lagerström, R., 2018. Load balancing of renewable energy: a cyber security analysis. Energy Inform. 1, 1–42. https://doi.org/10.1186/s42162-018-0010-x.

Wang, H., Jia, Z., Shen, Z., 2009. Research on security requirements engineering process. In: IE and EM 2009 - Proceedings 2009 IEEE 16th International Conference on Industrial Engineering and Engineering Management, pp. 1285–1288.

Withers, R., 2016. Software and attack centric integrated threat modeling for quantitative risk assessment. In: Proceedings of the Symposium and Bootcamp on the Science of Security.

Wuyts, K., Scandariato, R., Joosen, W., 2014. Empirical evaluation of a privacy-focused threat modeling methodology. J. Syst. Softw. 96, 122–138. https://doi.org/10.1016/j.jss.2014.05.075.

Wuyts, K., Sion, L., Joosen, W., 2020. Linddun go: a lightweight approach to privacy threat modeling. In: 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). IEEE, pp. 302–309.

Xiong, W., Lagerström, R., 2019. Threat modeling – a systematic literature review. Comput. Secur. 84. https://doi.org/10.1016/j.cose.2019.03.010.

Young, W., Leveson, N.G., 2014. Inside Risks an Integrated Approach to Safety and Security Based on Systems Theory, pp. 31–35.

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

Yu, E., Mylopoulos, J., 1998. Why goal-oriented requirements engineering. In: Proceedings of the 4th International Workshop on Requirements Engineering: Foundations of Software Quality, pp. 15–22.

Zahid, S., Mazhar, M.S., Abbas, S.G., Hanif, Z., Hina, S., Shah, G.A., 2023. Threat modeling in smart firefighting systems: aligning mitre att&ck matrix and nist security controls. Int. Things 22, 100766.

Zografopoulos, I., Konstantinou, C., Georgios Tsoutsos, N., Zhu, D., Broadwater, R., 2021a. Security Assessment and Impact Analysis of Cyberattacks in Integrated T&D Power Systems 21.

Zografopoulos, I., Ospina, J., Liu, X., Konstantinou, C., 2021b. Cyber-physical energy systems security: threat modeling, risk assessment, resources, metrics, and case studies. IEEE Access 9, 29775–29818. https://doi.org/10.1109/ACCESS.2021.3058403. arXiv:2101.10198.

Shaymaa Mamdouh Khalil is a PhD candidate and an early-stage researcher at the Center for Digital Forensics and Cyber Security at Tallinn University of Technology, Estonia. She holds a BSc in electronics and communication engineering from Cairo University (2006) and an MBA International Paris from the University Paris Dauphine and the University Paris 1 Pantheon Sorbonne (2013). In 2020, she 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.

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 Senior Researcher. He has published more than 20 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.

Appendix 2

II

H. Bahsi, H. O. Dola, S. M. Khalil, and T. Korõtko. A cyber attack taxonomy for microgrid systems. In *2022 17th Annual System of Systems Engineering Conference (SOSE)*, pages 324–331, 2022

A Cyber Attack Taxonomy for Microgrid Systems

Hayretdin Bahsi[1], Henry Ochieng' Dola[2], Shaymaa Mamdouh Khalil[3], and Tarmo Korõtko[4]

[1,2,3]School of Information Technology, Tallinn University of Technology
[4]Smart City Center of Excellence (Finest Twins), Tallinn University of Technology
hayretdin.bahsi, henry.dola, shaymaa.khalil, tarmo.korotko@taltech.ee

2022 17th Annual System of Systems Engineering Conference (SOSE) | 978-1-6654-9623-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/SOSE55472.2022.9812642

***Abstract*—Microgrid systems play a pivotal role in utilizing the distributed energy sources for reaching a more economic, safe, and environment-friendly energy ecosystem. A coherent adaptation of such systems requires a serious consideration of their resistance to cyber attacks due to the increasing threat profile of cyberspace and the potentially huge impact of malicious cyber actions on the energy systems. This study provides the results of our literature review regarding the cyber attacks on Microgrid systems. We created an attack taxonomy based on the review findings. This attack taxonomy can be utilized by practitioners for various purposes including security assessments of Microgrid systems or relevant cyber threat modeling studies.**

***Index Terms*—cyber attack; Microgrid; Taxonomy**

I. INTRODUCTION

The complexity of electricity transmission and distribution systems has been increasing with the penetration of distributed energy resources. The boundaries between proactive and reactive grid entities have become blurred and unidirectional energy flow is replaced with bidirectional flows, while consumers turn into prosumers [1]. Microgrid technology plays an eminent role in this new "distributed" ecosystem as it turns the localized entities into small- or medium-scale energy generation and distribution systems. Microgrids' capability of switching between islanded and connected modes makes them a feasible solution for being utilized in geographically distant locations or in crisis cases when long disruptions occur in the main grids.

As the Microgrid technology more benefits from computing and network technologies in such a highly critical energy sector, it is inevitable that malicious actors encompassing different profiles (e.g., hostile nation-states, criminal groups, terrorist organizations) may utilize cyber means to achieve their aims. The complexity of the technologies, rapid progress in technology adaptation and high demand for the realization of green energy policies may cause insecure systems which are susceptible to various types of cyber attacks. It is essential to understand the potential cyber threats and their implications in Microgrid systems.

The application of the Microgrid systems ranges from data centers, electrical vehicle charging stations to electric railway systems [2]. Cyber attacks on these systems may cause significant physical results regarding the disruption of the electricity supply. It also paves the way for the attackers to think in a multi-dimensional manner while designing attack scenarios, for instance, for conventional targets such as data centers. The disruption of the Microgrid system of such centers would be a more viable and effective option for launching denial of service attacks when compared to the common data-centric attacks in some situations.

Microgrid systems have a wide attack surface due to their considerably complex connectivity with many entities and high interaction with various users and administrators. For example, 6 different attack surfaces can be distinguished for the advanced metering infrastructure alone [3]. The day-to-day operations of such systems would be done by the owning entity, thus, network administrators and other operators frequently access the systems locally or remotely. There exist tertiary control relation with Distributes System Operators (DSOs), meaning that high connectivity is required between the DSO and Microgrid systems. Additionally, DSO operators may have remote access to some assets such as smart meters. As Microgrid Systems include many products from different vendors, the vendors need to do remote or local maintenance, creating another security burden. Insider threat poses a significant challenge as an insider can provide information regarding the physical and cyber layers or play more active roles in launching the attacks.

This study reviews the literature regarding the cyber attacks to Microgrid Systems and provides a taxonomy for the classification of identified attacks. An attack taxonomy is an essential instrument in threat modeling studies for achieving systematic threat elicitation. After modeling the system, the experts review the taxonomy and consider the applicability of the relevant attack categories for the corresponding system components. The output of this study will be utilized in the threat modeling study of a project that aims to establish a Microgrid system and develop management software for Microgrid operations. The proposed taxonomy can be utilized by experts in other threat modeling and security assessment projects.

While conducting the literature search, we selected a slightly broader scope so that, besides the studies specifically addressing Microgrid systems, we also covered the studies focusing on the attacks in the smart grid or similar energy systems. The reason for this choice is that the cyber security literature regarding Microgrid technology would be considered immature due to evolving nature of the domain. The cyber attacks threatening similar systems could be easily adapted to this specific technology as the underlying concepts and terms are so close.

The review part of this study does not claim to be system-

Authorized licensed use limited to: Tallinn University of Technology. Downloaded on March 30,2026 at 16:07:05 UTC from IEEE Xplore. Restrictions apply.

978-1-6654-9623-0/22/$31.00 ©2022 IEEE

atic. Although we contemplate that we have reached saturation to some extent in terms of identifying new terms within our scope, it has not been feasible to follow a systematic approach due to widely dispersed research communities (i.e., energy, cyber-physical systems, cyber security) and huge literature about the different cyber-physical domains which may not be captured with appropriate keywords. We preferred to be more pragmatic in identifying the studies by selecting different keywords and applying forward and backward snowballing techniques in an unsystematic way.

The structure of this paper is as follows: Section II gives the necessary background information about the system architecture and network topology of a Microgrid System and presents network protocols commonly used in this system. The attack taxonomies are reviewed in Section III. The cyber attack vectors and their classifications are given in Section IV. Section V concludes the paper.

II. Background Information

A. Microgrid System Architecture

Control structure in Microgrid systems is classified into three categories: (1) primary control (local controllers), (2) Secondary control and (3) Tertiary control [4], [5]. Local controls are conducted at the device level based on the local measurements. Distributed control system or energy management system managed within the Microgrid system acts as a secondary control whereas the control actions conducted by the upper-level of an organization such as distribution system operator (DSO) would be considered as tertiary control. This control level is mainly tasked with the economical operations and coordination of power exchange between DSOs and Microgrid systems [6]. The secondary/tertiary control levels need communication networks and are hierarchically implemented, meaning that the failure of the communication channels between the control centers and field devices (e.g., converters) would cause various malfunctions of the devices which may lead to cascading failures and system-level problems [6]. Such hierarchic relations may cause catastrophic security incidents as the compromise of the higher level may result in the compromise of the whole system at the lower levels. The denial of service attacks aiming to disrupt the communication channels may also have severe effects depending on the strictness of availability requirements or perfectness of attack timing.

In this section, we demonstrate a typical system architecture and network topology of a Microgrid system as shown in Figure 1. The system is mainly divided into two major networks: (1) Wide area network or neighborhood area network consisting of connections of Microgrid with DSO (i.e., large-scale Microgrids could be connected to TSO) and other Microgrids, (2) Local area network or home area network consisting of Microgrid control center, IED devices, smart meter, and router-modem/gateway. In this architecture, the controller center (i.e., functions as a secondary control layer) constitutes a sub-network that can be further divided into three separate network segments: (1) Server (2) Operator (3) Controller. The server segment hosts the following system components used for the secondary control of the whole Microgrid system: Energy management system (EMS), HMI server, and Historian database. The second network segment is dedicated to HMI consoles accessed by the operators. This segment can be composed of other sub-segments located in different parts of the facility depending on the size of the location and network infrastructure. The third segment hosts the controller servers which link the IED devices and EMS.

There exist three architecture levels in a typical substation network, station, bay, and process levels [7]. Mainly, the station level contains the SCADA components, the bay level hosts IED devices, and the process level includes sensors/actuators. Such architecture is composed of two network buses, a station bus and a process bus. The former links the station level and bay level elements whereas the latter provides direct network links between devices in bay and process levels. In our architecture, communication network corresponds to the station bus. Although we did not include the process bus in Figure 1, it could be added when needed.

B. Network Protocols

The incorporation of variable renewable sources of energy into the main energy grid will require a set of procedures to support their safe management and flexibility given electricity will flow to and from local producers and consumers [8]. For Microgrids specifically, there are several communication protocols based on grid applications [9], out of which we discuss the most commonly used.

1) IEEE 1815 (DNP3): IEEE 1815 (DNP3) is the communication protocol used for SCADA communications. Developed for supporting the utility industry, DNP3 is an open and optimized protocol developed for interactions between master and substations together with applications supporting their operations [10]. It is compatible with a host of topologies including multiple masters, multiple slaves, and peer-to-peer communications.

2) IEC 61850: IEC 61850 is the international information model and protocol for communications with Distributed Energy Resource (DER) and distribution automation. It constitutes the basis of various protocols used for distributing energy resources systems [11] and for the collection of information from a myriad of devices used in power systems. It is key in the relay of information between intelligent electronic devices (IED), the control center, and HMI leveraging on TCP/IP and MMS (Manufacturing Message Specification).

3) IEEE 2030.5 (SEP2): IEEE 2030.5 (SEP2) is a new protocol developed based on IEC 61850 for home area networks although it is used for other applications such as utility interactions with DER systems between smart grids and consumers. It is the application protocol for demand response, load control, pricing management of distributed generation and electric vehicles [12].

4) OPENADR: OPENADR is a protocol developed for demand response applications in the provision of pricing data [13]. Based on its interactive mode and transfer mechanism, it

Authorized licensed use limited to: Tallinn University of Technology. Downloaded on March 30,2026 at 16:07:05 UTC from IEEE Xplore. Restrictions apply.

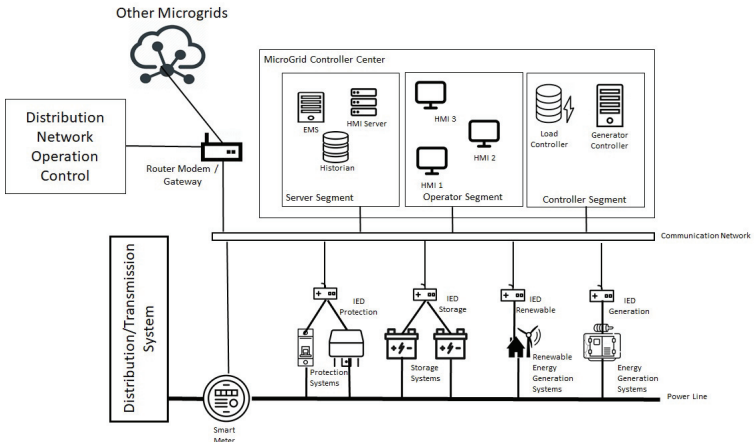


Fig. 1. Microgrid System Architecture

has been developed to aid utilities in cost-effectively managing the growing demand for electricity and the decentralized energy production between prosumers in the modernization of the electric grid.

III. Review of Attack Taxonomies

"ATT&CK for Industrial Control Systems" is a framework developed by MITRE for the classification and characterization of adversary behavior addressing industrial control systems [14]. This framework outlines the steps of adversary behavior in the form of various tactics and techniques. Although this framework presents a comprehensive taxonomy, the categories focus on defining the details of the adversarial actions. Threat modeling studies are usually done at the early stages of the system development so that detailed attack techniques would be more useful in the later development steps when the detailed system design is known. Therefore, we contemplate that MITRE's framework does not suit our purpose exactly.

A very high-level attack classification is given in [15] for SCADA systems. Although the definitions are not clear, the study categorizes the attacks on SCADA systems as hardware, software, and communication attacks. Communication attacks are further split into sub-categories such as network, transport, application and protocol. Another study provides taxonomies for attacks, vulnerabilities, and damage of the attacks that can be applied to control systems in the energy sector [16]. An incident classification taxonomy for CSIRT teams of smart grids has been proposed in [17]. As the main idea is to handle the incidents, the systems targeted by the attacks are considered in the classification due to the requirements for dividing the incident handling roles among different IT and Industrial control system teams.

IV. Attack Vectors and Their Classification

As shown in Figure 2, we have identified five major categories: reconnaissance, device compromise, network compromise, integrity attacks, and denial of service attacks. It is a typical choice to analyze cyber attacks under the common categories, confidentiality, integrity, and denial of service attacks. However, in most cases, reconnaissance and device/network compromise activities, which have their own technical approaches and procedures, constitute prerequisites for all the attacks regardless of their actual impacts. We did not create a specific category for the confidentiality attacks as device or network compromise may enable the attacker to access the data or reconnaissance activities aim to reveal critical information regarding the target system, meaning that such attacks are intrinsically represented in those categories. Although some attacks such as eavesdropping could be considered a confidentiality attack that could be categorized under a distinct category, we prefer a more simplistic categorization approach for better comprehension and listed them under network or device compromise categories.

Each category is further divided into sub-categories which are mostly defined based on the variations of the attack objective or technique. For instance, if the objective of the availability attack is to manipulate the routing communication, we considered that that attack belongs to "manipulation of routing protocols". On the other side, if the attack addresses the communication about time data exchange, it belongs to the "attacks to time synchronisation" category. It is important to point out that each sub-category may not be mutually exclusive. For instance, a spoofing attack would be a part or prerequisite of a man-in-the-middle attack. We briefly explained each attack category and sub-category in the remaining part of this section.

Authorized licensed use limited to: Tallinn University of Technology. Downloaded on March 30,2026 at 16:07:05 UTC from IEEE Xplore. Restrictions apply.

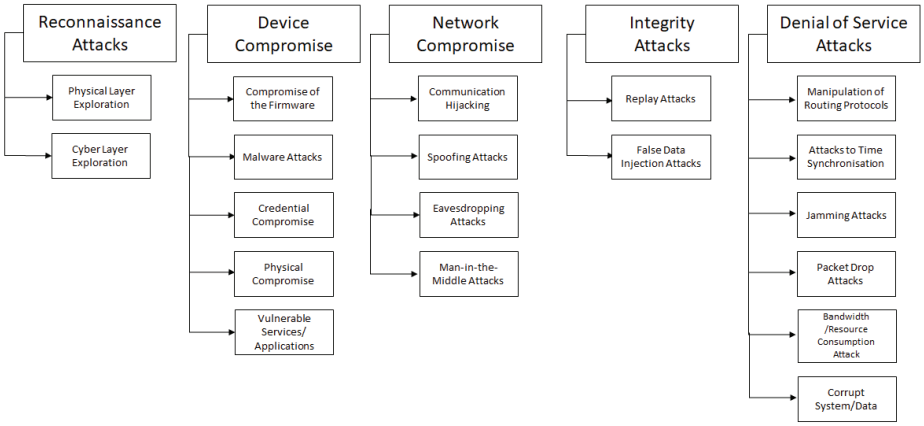


Fig. 2. The Classification of Attack Vectors

A. Reconnaissance Attacks

A malicious actor always needs detailed information about the target system in order to prepare for the attack. While addressing a cyber-physical system such as a Microgrid system, the attacker should devote more effort when compared to traditional network security attacks as he should focus on both physical and cyber layers. The amount of required information is mostly proportional to the required sophistication level of the attack and can be obtained by using various technical or social means.

1) Physical Layer Exploration: The physical topology of the targeted power system constitutes a significant input for FDI attacks [18]. Past intruders or former utility company employees can provide topology data, which is termed an offline data collection method, whereas the attackers can employ online methods such as the deployment of their own sensors and PMUs [19]. Locational marginal prices (LMPs) obtained from publicly available market data would be another data source for revealing the target topology [20]. It is also demonstrated that the attackers can deduce the topology information from power flow measurements, thus, just having such measurement data would be enough to launch an FDI attack [21]. Publicly shared information in the energy sector such as newsletters, success stories, or reports released by utility companies can be important sources for the attackers [22].

2) Cyber Layer Exploration: Obtaining data about the network topology in the cyber layer is a well-known practice in the cyber security domain. The attacker can resort to passive methods such as collecting data from cyber threat intelligence resources such as Shodan (i.e., a website providing OS details of devices connected to the Internet) or active methods by using network and security scanners horizontally and vertically.

B. Device Compromise

1) Compromise of The Firmware: The firmware of embedded devices could be compromised as a prerequisite for launching other attacks such as integrity and denial of service attacks as described in Sections IV-D and IV-E. Embedded devices usually perform code execution in single-threaded loops without requiring an operating system [23]. The attackers can target the integrity of the code execution and take control of the devices (e.g., inverter controllers). The prerequisites for conducting malicious code execution would be to physically access the device (i.e., some devices may be located in places where physical security is not guaranteed), supply-chain type of attacks, or compromising the device maintenance process (e.g., compromising the vendor's codebase, bribing the maintenance staff or other insider threat vectors).

A case study demonstrating such an attack and its results in a Microgrid system is given in [23]. Specifically, the maximum power point tracking controller of an inverter of a solar PV is compromised by the modification of its firmware code. The impact of the attack is evaluated according to the frequency and voltage stability metrics.

2) Physical Compromise: An adversary may resort to physical means to access the devices in the field (e.g., sensors, actuators, IEDs) and replace them with the ones they can remotely control or deploy malware to the devices. Although this attack necessitates to know the details of ongoing processes, an adversary can launch this attack after collecting the relevant data by using the reconnaissance methods given in Section IV-A. Depending on the capabilities and opportunities of the attacker, it is possible that the firmware of an existing device can be replaced with a malicious one after physically capturing the device itself.

In some attack scenarios, the attacker may be interested to deploy an additional rogue device instead of replacing an

Authorized licensed use limited to: Tallinn University of Technology. Downloaded on March 30,2026 at 16:07:05 UTC from IEEE Xplore. Restrictions apply.

existing one and use that device as a foothold for further attacks. This device can be controlled remotely via wireless or 5G modem interfaces.

3) Malware Attacks: Botnets consist of bot computers that are controlled by an attacker group to launch cyber attacks (e.g., denial of service, spear-phishing, spamming) on their main targets [24]. In a typical botnet attack scenario, the attackers consume the bot resources such as bandwidth and CPU for their purpose and also hide themselves behind the innocent computers, making attack attribution so difficult.

Although botnet is a well-known and considerably well-studied topic in the cyber security field, the main focus has been always on the cyber attacks originating from bots. However, it is possible that the attackers can compromise IoT devices to initiate large-scale physical consequences rather than create information-centric harm. In a simulation study, it is demonstrated that a botnet composed of high wattage IoT devices can be utilized by the attackers to create sudden energy consumption changes which may lead to significant disruption via frequency instabilities or line failures in the whole power grid [25]. Although we have not identified a study that demonstrates how a compromise of various Microgrids can cause such a wider disruption, we argue that a malicious actor may consider this option as the potential vulnerabilities in the commonly-utilized Microgrid products may enable the creation of relatively small but effective botnets.

4) Credential Compromise: The malicious actors may access the target devices physically or remotely and aim to obtain user or admin credentials. In case of achieving to get only credentials of a restricted user, the actor may benefit from local vulnerabilities to get admin rights. The defaults passwords set by the vendors constitute an important security problem. The attacker can also resort to password cracking attacks (e.g., brute force, guessing).

5) Vulnerable Services and Applications: The vulnerabilities of the network services or applications that can be accessed remotely are usually utilized by the attackers for compromising the system assets. When the devices are accessed physically, the local vulnerabilities of the running applications can be also compromised for having full control of the victim.

C. Network Compromise

1) Communication Hijacking: This attack vector enables the adversary to take control of communication between different entities in the network. Although the MiM attack vector, which has been addressed as a different attack vector in this report, can play a key role in the compromise of the communication channels, we prefer to create a more general category under this term as the attackers can resort to other types of attacks. Nevertheless, this attack vector necessitates the malicious actor to get a foothold in the target network by placing a rogue device or compromising an existing one.

As GOOSE protocol within IEC 61850 standard employs a multicast approach based on publish/subscriber model, hijacking a communication does not require a man-in-the-middle type of attack typically performed for unicast protocols. After joining the group of the corresponding multicast network, a malicious node obtains the targeted GOOSE messages and resubmits similar messages with a high-status number or floods the network with messages having increasing status numbers so that the modified messages rather than the original ones would be processed by the subscribers [26].

2) Spoofing Attacks: The industrial control protocols such as DNP3 and IEC 61850 do not provide the authentication of the communicating entities and the messages. Therefore, once an attacker gets a foothold in the relevant network (i.e., process, bay, and station) it is easy to spoof the MAC address of any target device and conduct network layer 2 attacks. Publish/Subscribe communication model applied by IEC 61850-8-1 makes the attack easier as the attacker can easily assume the role of publisher or subscriber and further conduct other attacks. A practical spoofing attack with the GOOSE protocol of IEC 61850 is demonstrated in [27]. Although a security standard IEC 62351-6 introduces authentication options for the GOOSE messages, those countermeasures can not be applied in cases where high real-time requirements (e.g., 4ms response time) should be provided [28].

3) Eavesdropping: This is a passive attack type in which an adversary listens to the communication channel (i.e., wireless or wired) and obtains the messages between nodes. Although the attack is passive and aims to compromise the confidentiality property, it may enable the attacker to use the data in the later phases of an attack scenario for integrity or availability attacks.

The encryption methods provide a countermeasure against eavesdropping devices. Still, the adversary can try to steal crypto credentials and use them for decrypting the messages or apply various cryptoanalysis techniques. On the other side, in case of proper application of encryption methods, the adversary can apply traffic analysis techniques to identify the communication patterns if needed in the attack scenario. A traffic analysis case for PMU is discussed in [29].

An eavesdropping case in an advanced metering infrastructure (AMI) is briefly captured in [30] besides other types of attacks. The eavesdropping of smart meter communication using DL/T 64-2007 is demonstrated in [31].

4) Man-in-the-Middle (MiM) Attacks: This refers to a category of impersonation attacks where an intruder embeds himself between two devices to either impersonate or eavesdrop on the devices. The intruder intercepts and relays private communication between victims in real-time creating an impression they are directly communicating with each other [32], deemed a masquerade attack, while session hijackers on the other hand intercept meter connection seeking to steal, modify or delete transmitted data. An attacker could then use the gained information to conduct malicious activities such as tampering information transferred in packets (e.g., pricing details).

Sample attacks are simulated on a testbed with an electricity distribution operator [33] to show the impact of MiM attacks on SCADA servers. A similar scenario is realized between two

Authorized licensed use limited to: Tallinn University of Technology. Downloaded on March 30,2026 at 16:07:05 UTC from IEEE Xplore. Restrictions apply.

legitimate devices, depicting the interception and modification of packets as demonstrated in [34].

D. Integrity Attacks

These attacks violate the integrity properties of control or measurement data to induce behavioral changes in sensors or actuators for achieving disruption or destruction purposes [35].

1) Replay Attacks: In this attack, the malicious actors aim to harm the target energy function by collecting control and measurement data and then resending them at the relevant time, meaning that the modified part is a temporal aspect rather than the data itself [36]. Replay attacks may not require much knowledge about the system components [37] but the attackers should be able to obtain the data by compromising the targeted communication channels.

2) False Data Injection: This attack category targets the integrity of measurement data gathered from the sensors or control data among different control layers in order to create disruptive harm to various critical energy functions such as state estimation, voltage and frequency control, and system protection [38]. Although the measurement errors would be identified by bad data detection mechanisms in power systems, it is shown that maliciously introduced errors may go unnoticed [39]. As a prerequisite, attackers should have partial or complete network topology information and details of the processes. Additionally, they should be able to manipulate the measurement data before it arrives at the control center where the computation for critical decisions (e.g., state estimation) is performed [39]. As this attack requires more hidden activities as opposed to, for instance, denial of service attacks, it is also characterized as deceptive [40] or stealthy and can be a relevant attack vector for advanced persistent threats. In more sophisticated forms of these attacks, stronger stealthiness assumptions are held. For instance, a successful attack can be designed so that the power system state estimator converges after the data modification, or instead of complete information about the target system model, a perturbed or outdated model would suffice [41].

It is demonstrated that it is still possible to launch this attack with a limited number of compromised measurement devices (i.e, a compromise of 10 meters among 1.122 meters in the IEEE 300-bus system would be enough) [39]. The attacks that can be carried out with limited compromised devices are called sparse FDI attacks [42]. On the other side, there exist studies that show an FDI attack is possible even if the attackers do not have any grid knowledge but they can still predict the topology by correlating the measurement data (i.e., these attacks are named blind stealthy attacks) [42].

FDI attacks can be also classified based on the approach to how the false measurement is chosen. In the scaling attack, the higher and lower values are determined as the false data whereas the selection of a random value from a uniform function is named random attack [43]. Some approaches incorporate time information into the value selection procedure as ramp attacks gradually increase or decrease the value, and pulse attacks insert data in specific time intervals [43].

The impact of an FDI attack that alters the frequency stability of the isolated Microgrid system is analyzed in [44]. The simulation of an exogenous attack (i.e., modifying the frequency deviation with additive value) provides a solid impact on the Microgrid state whereas an endogenous attack (i.e., manipulating the frequency with multiplication parameter) caused more varying and obscure results.

The control data could be also the subject of a false data injection attack. A case study shows that the manipulation of load demand control data at various devices can disturb the frequency stability [23].

Another interesting variation of false injection attack aims to send modified price data to smart meters (i.e., from a source on the Internet or DSO) so that household appliances utilize more energy assuming to be in the period of low price [45]. If the attacker targets many customers with this attack vector at the same time, this coordinated action may induce a sudden increase in energy demand, thus, major failures in the main grid.

A comprehensive survey about FDI attacks is given in [42]. This study classifies these attacks according to three aspects, their attack models, target systems, or attack impact. Attack models consider the flow model of the target system (i.e., whether it is a DC or AC power model), the network control structure (i.e., distributed or centralized), and the construction method. The last notion mostly analyses the attacks from the perspective of required knowledge such as the amount of topology information (i.e., complete, partial, or none). In our study, we do not further split this attack into subcategories in order not to make the classification more complicated as we cover all types of cyber attacks in our scope.

E. Denial of Service Attacks

Denial of service (DOS) attacks earmark the core component of an energy grid i.e. availability which is ensured using intelligent electronic devices [46] for the exchange of consumer data, updates and electricity rates from utility companies. The electricity grid is susceptible to attacks channeled through communication infrastructure used for data transmission and control.

These kinds of attacks can be executed through several means, no longer solely deliverable by the physical means but a compromised node in the target system (e.g., via malware or exploitation of the hardware or software security flaws) may be utilized as a foothold for achieving the malicious purpose. The sub-categories of these attacks are given below.

1) The Manipulation of the Routing Protocols: Smart Grid systems, in general, and Microgrid systems, in particular, may benefit from various network types including wireless mesh topologies [12], which run considerably complex routing protocols (e.g., dynamic source routing [47]). A malicious actor may target the routing traffic instead of the actual data communication to disrupt the connectivity in such dynamically-formed networks. A sample attack, namely puppet attack, and its impact are analyzed in an advanced metering infrastructure [48]. The attack method is based on the idea that a malicious

Authorized licensed use limited to: Tallinn University of Technology. Downloaded on March 30,2026 at 16:07:05 UTC from IEEE Xplore. Restrictions apply.

node in the target network may induce excessive routing traffic by creating and advertising modified routing data packets. The attacker can also conduct other malicious actions such as re-routing the packets to non-existent routers or simply dropping the packets [49]. The malicious actor should take control of an existing device with the routing function or place a masqueraded external device which participates in routing data exchanges in the network.

2) Attacks to Time Synchronization: Smart grid systems depend on strict real-time requirements, necessitating accurate and prompt management of the time data. The attacks on time synchronization may cause delays in the transmission of the time data or end up with serious integrity violations on the timestamp values of the measurement data.

IEEE 1588 Precision Time Synchronization Protocol (PTP), based on a master-slave model, is proposed for power grid systems to do synchronization in the range of nanoseconds [50]. In this protocol, master and slave nodes exchange information to agree on the same time value. It is demonstrated that an attacker can conduct a man-in-the-middle type of attack to master and slave nodes, intercept the messages, modify them (e.g., introducing additional delays) and induce a disruption in time synchronization [50].

GPS is also utilized in various power grid systems. It is shown that a GPS spoofing attack against PMUs (Phasor Measurement Units) can lead to time manipulations on the phase angle measurement data, posing a significant threat to grid stability [51].

3) Jamming Attacks: This is a class of DOS attacks on cyber-physical system applications which use the physical communication layer of wireless protocols as an attack medium, and the performance of the networks can be degraded or blocked through the emission of an interference signal thereby disrupting control [52]. An attack on the electric grid could have disastrous effects [53] i.e. if the sensors in communication with a remotely located estimator (IED) would have their channel jammed thereby inhibiting interaction.

4) Packet Drop Attack: In this attack, an adversary compromises a node (i.e., via physical or cyber means), a gateway, or routing device in the target network and intentionally drops or delays the network communication passing over that compromised node. Packets dropped en masse can easily be noticed [54]. As such a successful packet drop attack would result in only the loss of a subset of the packets although the adversary would aim to maximize damage without being noticed [54].

5) Bandwidth or Resource Consumption Attack: Filling the network bandwidth of the target system with excessive communication requests is one of the common DOS attack types which can be also applicable to Microgrid Systems. Instead of addressing network bandwidth, some attack types may aim to overuse the local resources of the devices (e.g., RAM, CPU, Operating system limits) by sending excessive requests or crafted network packets.

6) System or Data Corruption: The attacker may aim to corrupt the OS, firmware, or data of the target system once accessing it physically or remotely. Corrupting the boot loader, replacing the firmware with a non-functional one and deleting critical OS or other data files are some attack vectors that can be applied when the malicious actor has the opportunity to access the target with some privileges.

V. Conclusion

In this study, we reviewed the literature for identifying the potential cyber attack vectors that may target Microgrid systems and then provided an attack taxonomy based on the review findings. The taxonomy can be utilized in cyber threat modeling studies or security assessments of Microgrid systems.

It is important to note that having a complete set of attack vectors is not practically possible as the target system is complex and there exists a huge number of attack methods that can not be completely listed. Some of the methods are not mutually exclusive. However, we contemplate that our taxonomy reflects a significant portion of the relevant knowledge in the literature.

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

[1] T. Korõtko, A. Rosin, and R. Ahmadiahangar, "Development of prosumer logical structure and object modeling," in *2019 IEEE 13th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG)*, 2019, pp. 1–6.

[2] F. Nejabatkhah, Y. W. Li, and H. Tian, "Power quality control of smart hybrid ac/dc microgrids: An overview," *IEEE Access*, vol. 7, pp. 52 295–52 318, 2019.

[3] A. Antonov, T. Häring, T. Korõtko, A. Rosin, T. Kerikmäe, and H. Biechl, "Pitfalls of machine learning methods in smart grids: A legal perspective," in *2021 International Symposium on Computer Science and Intelligent Controls (ISCSIC)*, 2021, pp. 248–256.

[4] D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi, C. A. Cañizares, R. Iravani, M. Kazerani, A. H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke *et al.*, "Trends in microgrid control," *IEEE Transactions on smart grid*, vol. 5, no. 4, pp. 1905–1919, 2014.

[5] R. AhmadiAhangar, A. Rosin, A. N. Niaki, I. Palu, and T. Korõtko, "A review on real-time simulation and analysis methods of microgrids," vol. 29, no. 11, 2019, p. e12106.

[6] V. Nasirian, S. Moayedi, A. Davoudi, and F. L. Lewis, "Distributed cooperative control of dc microgrids," *IEEE Transactions on Power Electronics*, vol. 30, no. 4, pp. 2288–2303, 2014.

[7] S. E. Quincozes, C. Albuquerque, D. Passos, and D. Mossé, "A survey on intrusion detection and prevention systems in digital substations," *Computer Networks*, vol. 184, p. 107679, 2021.

[8] Y. Yang, T. Littler, S. Sezer, K. McLaughlin, and H. Wang, "Impact of cyber-security issues on smart grid," in *2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies*. IEEE, 2011, pp. 1–7.

[9] M. Z. Gunduz and R. Das, "Analysis of cyber-attacks on smart grid applications," in *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*. IEEE, 2018, pp. 1–5.

[10] A. M. Khattak, S. I. Khanji, and W. A. Khan, "Smart meter security: Vulnerabilities, threat impacts, and countermeasures," in *International Conference on Ubiquitous Information Management and Communication*. Springer, 2019, pp. 554–562.

[11] J. Bergmann, C. Glomb, J. Götz, J. Heuer, R. Kuntschke, and M. Winter, "Scalability of smart grid protocols: Protocols and their simulative evaluation for massively distributed ders," in *2010 First IEEE International Conference on Smart Grid Communications*. IEEE, 2010, pp. 131–136.

Authorized licensed use limited to: Tallinn University of Technology. Downloaded on March 30,2026 at 16:07:05 UTC from IEEE Xplore. Restrictions apply.

[12] S. Kumar, S. Islam, and A. Jolfaei, "'microgrid communications—protocols and standards," *Variability, Scalability and Stability of Microgrids*, vol. 139, p. 291, 2019.

[13] G. Clarke, D. Reynders, and E. Wright, "Preview of dnp3," in *Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems*. Burlington, MA, 2004.

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

[15] B. Zhu, A. Joseph, and S. Sastry, "A taxonomy of cyber attacks on scada systems," in *2011 International conference on internet of things and 4th international conference on cyber, physical and social computing*. IEEE, 2011, pp. 380–388.

[16] T. Fleury, H. Khurana, and V. Welch, "Towards a taxonomy of attacks against energy control systems," in *International Conference on Critical Infrastructure Protection*. Springer, 2008, pp. 71–85.

[17] R. de Jesus Martins, L. A. D. Knob, E. G. da Silva, J. A. Wickboldt, A. Schaeffer-Filho, and L. Z. Granville, "Specialized csirt for incident response management in smart grids," *Journal of Network and Systems Management*, vol. 27, no. 1, pp. 269–285, 2019.

[18] G. Liang, J. Zhao, F. Luo, S. R. Weller, and Z. Y. Dong, "A review of false data injection attacks against modern power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1630–1638, 2016.

[19] M. A. Rahman and H. Mohsenian-Rad, "False data injection attacks with incomplete information against smart power grids," in *2012 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2012, pp. 3153–3158.

[20] V. Kekatos, G. B. Giannakis, and R. Baldick, "Grid topology identification using electricity prices," in *2014 IEEE PES general meeting—conference & exposition*. IEEE, 2014, pp. 1–5.

[21] M. Esmalifalak, H. Nguyen, R. Zheng, and Z. Han, "Stealth false data injection using independent component analysis in smart grid," in *2011 IEEE international conference on smart grid communications (SmartGridComm)*. IEEE, 2011, pp. 244–248.

[22] C. Konstantinou, M. Sazos, and M. Maniatakos, "Attacking the smart grid using public information," in *2016 17th Latin-American Test Symposium (LATS)*. IEEE, 2016, pp. 105–110.

[23] I. Zografopoulos, J. Ospina, X. Liu, and C. Konstantinou, "Cyber-physical energy systems security: Threat modeling, risk assessment, resources, metrics, and case studies," *IEEE Access*, vol. 9, pp. 29 775–29 818, 2021.

[24] J. Leonard, S. Xu, and R. Sandhu, "A framework for understanding botnets," in *2009 International Conference on Availability, Reliability and Security*. IEEE, 2009, pp. 917–922.

[25] S. Soltan, P. Mittal, and H. V. Poor, "Blackiot: Iot botnet of high wattage devices can disrupt the power grid," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 15–32.

[26] N. S. Kush, E. Ahmed, M. Branagan, and E. Foo, "Poisoned goose: Exploiting the goose protocol," in *Proceedings of the Twelfth Australasian Information Security Conference (AISC 2014)[Conferences in Research and Practice in Information Technology, Volume 149]*. Australian Computer Society Inc., 2014, pp. 17–22.

[27] J. Hoyos, M. Dehus, and T. X. Brown, "Exploiting the goose protocol: A practical attack on cyber-infrastructure," in *2012 IEEE Globecom Workshops*. IEEE, 2012, pp. 1508–1513.

[28] R. Schlegel, S. Obermeier, and J. Schneider, "A security evaluation of iec 62351," *Journal of Information Security and Applications*, vol. 34, pp. 197–204, 2017.

[29] B. Sikdar and J. H. Chow, "Defending synchrophasor data networks against traffic analysis attacks," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 819–826, 2011.

[30] D. Grochocki, J. H. Huh, R. Berthier, R. Bobba, W. H. Sanders, A. A. Cárdenas, and J. G. Jetcheva, "Ami threats, intrusion detection requirements and deployment recommendations," in *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2012, pp. 395–400.

[31] W. Dong, Y. Wang, L. Zhou, K. Cao, and H. Li, "An anti-eavesdropping method in data collection of smart meter," *Journal of Computer and Communications*, vol. 7, no. 9, pp. 38–49, 2019.

[32] P. Wlazlo, A. Sahu, Z. Mao, H. Huang, A. Goulart, K. Davis, and S. Zonouz, "Man-in-the-middle attacks and defense in a power system cyber-physical testbed," *arXiv preprint arXiv:2102.11455*, 2021.

[33] P. Maynard, K. McLaughlin, and B. Haberler, "Towards understanding man-in-the-middle attacks on iec 60870-5-104 scada networks," in *2nd International Symposium for ICS & SCADA Cyber Security Research 2014 (ICS-CSR 2014) 2*, 2014, pp. 30–42.

[34] I. Darwish, O. Igbe, O. Celebi, T. Saadawi, and J. Soryal, "Smart grid dnp3 vulnerability analysis and experimentation," in *2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing*. IEEE, 2015, pp. 141–147.

[35] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE transactions on automatic control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[36] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *2009 47th annual Allerton conference on communication, control, and computing (Allerton)*. IEEE, 2009, pp. 911–918.

[37] A. Teixeira, D. Pérez, H. Sandberg, and K. H. Johansson, "Attack models and scenarios for networked control systems," in *Proceedings of the 1st international conference on High Confidence Networked Systems*, 2012, pp. 55–64.

[38] F. Nejabatkhah, Y. W. Li, H. Liang, and R. R. Ahrabi, "Cyber-security of smart microgrids: A survey," *Energies*, vol. 14, no. 1, p. 27, 2021.

[39] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, pp. 1–33, 2011.

[40] X.-K. Liu, C. Wen, Q. Xu, and Y.-W. Wang, "Resilient control and analysis for dc microgrid system under dos and impulsive fdi attacks," *IEEE Transactions on Smart Grid*, 2021.

[41] A. Teixeira, S. Amin, H. Sandberg, K. H. Johansson, and S. S. Sastry, "Cyber security analysis of state estimators in electric power systems," in *49th IEEE conference on decision and control (CDC)*. IEEE, 2010, pp. 5991–5998.

[42] H. T. Reda, A. Anwar, and A. Mahmood, "Comprehensive survey and taxonomies of false injection attacks in smart grid: Attack models, targets, and impacts," *arXiv preprint arXiv:2103.10594*, 2021.

[43] S. Sridhar and M. Govindarasu, "Model-based attack detection and mitigation for automatic generation control," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 580–591, 2014.

[44] A. O. Aluko, R. P. Carpanen, D. G. Dorrell, and E. E. Ojo, "Vulnerability analysis of false data injection attacks on the frequency stability of isolated microgrids," in *2021 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)*. IEEE, 2021, pp. 1–6.

[45] S. Mishra, X. Li, T. Pan, A. Kuhnle, M. T. Thai, and J. Seo, "Price modification attack and protection scheme in smart grid," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1864–1875, 2016.

[46] P. Yi, T. Zhu, Q. Zhang, Y. Wu, and J. Li, "A denial of service attack in advanced metering infrastructure network," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 1029–1034.

[47] D. Johnson, Y.-c. Hu, D. Maltz *et al.*, "The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4," RFC 4728, Tech. Rep., 2007.

[48] P. Yi, T. Zhu, Q. Zhang, Y. Wu, and L. Pan, "Puppet attack: A denial of service attack in advanced metering infrastructure network," *Journal of Network and Computer Applications*, vol. 59, pp. 325–332, 2016.

[49] T. Zhu, S. Xiao, Y. Ping, D. Towsley, and W. Gong, "A secure energy routing mechanism for sharing renewable energy in smart microgrid," in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2011, pp. 143–148.

[50] Q. Yang, D. An, and W. Yu, "On time desynchronization attack against ieee 1588 protocol in power grid systems," in *2013 IEEE Energytech*. IEEE, 2013, pp. 1–5.

[51] D. P. Shepard, T. E. Humphreys, and A. A. Fansler, "Evaluation of the vulnerability of phasor measurement units to gps spoofing attacks," *International Journal of Critical Infrastructure Protection*, vol. 5, no. 3-4, pp. 146–153, 2012.

[52] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *IEEE Communications surveys & tutorials*, vol. 13, no. 2, pp. 245–257, 2010.

[53] Z. A. Baig and A.-R. Amoudi, "An analysis of smart grid attacks and countermeasures," *Journal of Communications*, vol. 8, no. 8, pp. 473–479, 2013.

[54] X. Zhang, S. F. Wu, Z. Fu, and T.-L. Wu, "Malicious packet dropping: how it might impact the tcp performance and how we can detect it," in *Proceedings 2000 International Conference on Network Protocols*. IEEE, 2000, pp. 263–272.

Authorized licensed use limited to: Tallinn University of Technology. Downloaded on March 30,2026 at 16:07:05 UTC from IEEE Xplore. Restrictions apply.

Appendix 3

III

S. M. Khalil, H. Bahsi, H. O. Dola, T. Korõtko, K. McLaughlin, and V. Kotkas. Threat modeling of cyber-physical systems - a case study of a microgrid system. *Computers & Security*, 124:102950, 2023

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.

© 2022 The Author(s). Published by Elsevier Ltd.
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

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
0167-4048/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

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

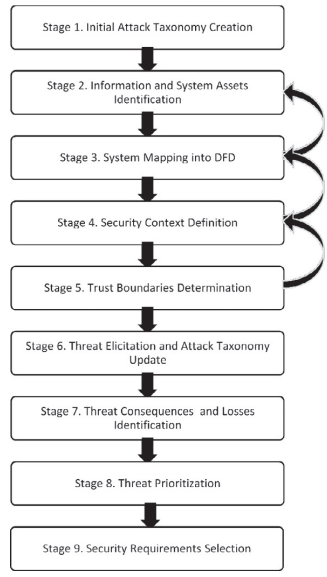


Fig. 1. Main Stages of the Proposed CPS Threat Modeling Methodology.

[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

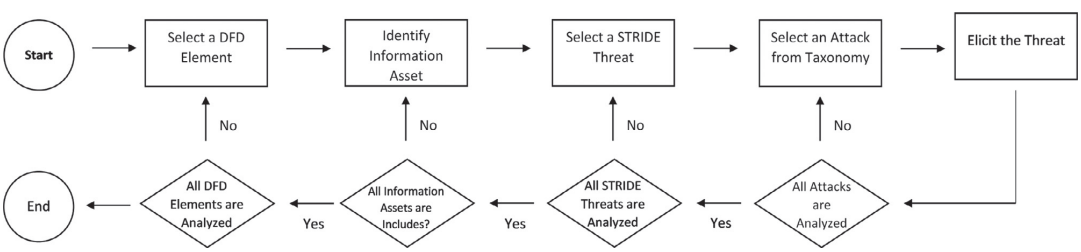


Fig. 2. Threat Elicitation Procedures.

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

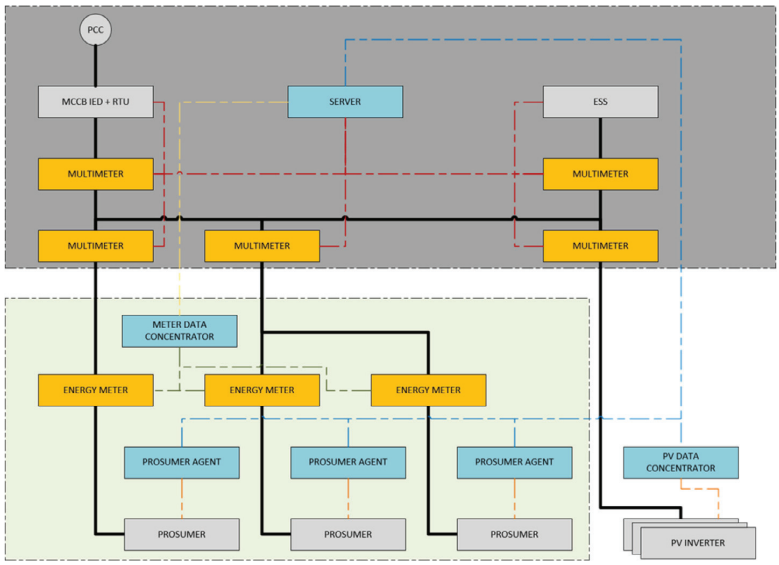


Fig. 3. Microgrid Initial Block Diagram.

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

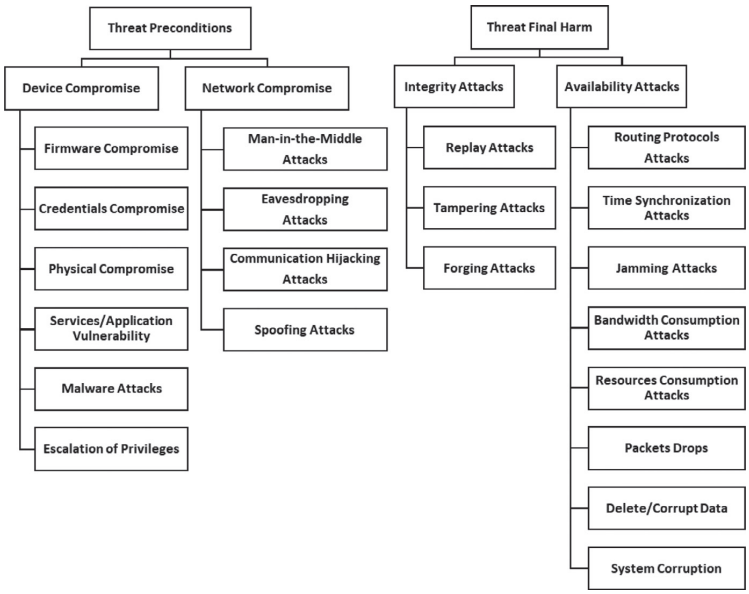


Fig. 4. Proposed Attack Taxonomy for Microgrid.

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

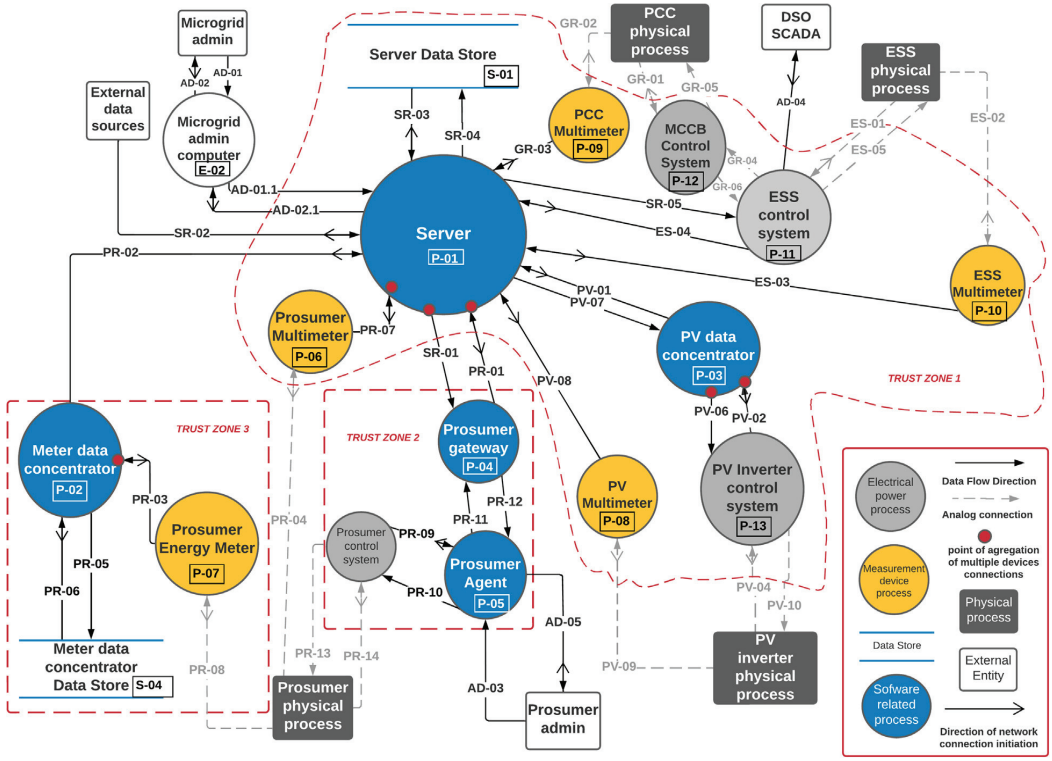


Fig. 5. Microgrid Data Flow Diagram.

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.

Appendix 4

IV

S. M. Khalil, H. Bahsi, and T. Korõtko. Penetration testing of newly-deployed industrial control systems. In J. Staggs and S. Shenoi, editors, *Critical Infrastructure Protection XIX: 19th IFIP WG 11.10 International Conference, ICCIP 2025, Revised Selected Papers*, volume 763 of *IFIP Advances in Information and Communication Technology*, pages 25–48, Cham, 2026. Springer



Penetration Testing of Newly-Deployed Industrial Control Systems

Shaymaa Mamdouh Khalil[1(✉)], Hayretdin Bahsi[1,2], and Tarmo Korotko[1]

[1] Tallinn University of Technology, Tallinn, Estonia
shaymaa.khalil@taltech.ee
[2] Northern Arizona University, Flagstaff, AZ, USA

Abstract. Penetration testing is a security practice that proactively discovers and assesses the potential exploitation of system vulnerabilities by simulating real-world attacks. Penetration testing of industrial control systems is a complex, time-consuming and demanding process requiring expertise in information technology and operational technology systems. Effective penetration test planning is crucial to prioritizing activities and optimizing time and resources.

This chapter describes a seven-stage methodology for penetration testing newly-deployed industrial control systems and demonstrates its practical application using a real-world case study of a newly-designed microgrid system. The methodology leverages threat modeling outcomes from the design phase to define the penetration testing scope and guide test choices, leading to an efficient testing process. Furthermore, it engages a novel MITRE ATT&CK to STRIDE mapping, which bridges the design phase that focuses on high-level threats and the planning phase that employs medium-level granularity. The methodology benefits security practitioners by enhancing planning efficiency and streamlining collaboration with third-party penetration testers. The methodology in conjunction with the MITRE ATT&CK to STRIDE mapping can be used to develop detailed, effective and efficient industrial control system penetration testing efforts.

Keywords: Threat Modeling · Industrial Control System · Penetration Testing · MITRE ATT&CK · STRIDE · Microgrid

1 Introduction

Industrial control systems are complex systems of systems that rely on information technology and operational technology. Due to their essential role in critical infrastructure assets, securing industrial control systems from cyber attacks is a priority. Following the Stuxnet incident, the application of secure-by-design principles has become vital to industrial control systems across the global critical infrastructure.

Threat modeling is an analytical process that identifies potential threats to systems and supports the selection of security requirements during the early stages of the system development lifecycle [9]. As expected, threat modeling

© IFIP International Federation for Information Processing 2026
Published by Springer Nature Switzerland AG 2026
J. Staggs and S. Shenoi (Eds.): ICCIP 2025, IFIP AICT 763, pp. 25–48, 2026.
https://doi.org/10.1007/978-3-032-11914-8_2

plays a vital role in ensuring the secure-by-design development of industrial control systems. Several threat modeling methods have been proposed to address the complex nature of industrial control systems [8]. However, a gap exists with regard to readily-deployable and well-validated industrial control system threat modeling methods. Validating the outcomes of threat modeling exercises and ensuring that the security requirements have been applied effectively to implemented systems are significant challenges.

The IEC 62443 standard [7] stipulates practices for a secure industrial control system development lifecycle. The practices include the specification of security requirements (using threat modeling), application of secure-by-design concepts (including how the design mitigates the identified threats), and security verification and validation testing. The standard identifies four processes related to security verification and validation testing that should be executed independently of system developers – security requirements testing, threat mitigation testing, vulnerability testing and penetration testing. However, existing guidelines do not provide detailed recommendations on how testing efforts can be harmonized.

Security requirements testing, as outlined in IEC 62443, ensures that system security functions satisfy the security requirements. This involves several testing approaches, including functional testing, performance testing and unexpected input testing, to assess system behavior under various scenarios.

Threat mitigation testing, on the other hand, verifies the effectiveness of deployed threat mitigation techniques. It also validates that the security measures address the threats identified during the threat modeling phase. For example, various authentication tests may be performed to ensure that an implemented authentication mechanism effectively mitigates spoofing threats.

Vulnerability assessments focus on identifying system security weaknesses. Unlike penetration testing, vulnerability assessments do not typically involve the exploitation of vulnerabilities. However, they serve as a foundation for penetration testing activities.

Penetration testing, by contrast, adopts a proactive approach. It simulates real-world cyber attacks by attempting to exploit vulnerabilities in order to assess the overall security postures of systems. It pushes organizations to recognize and proactively address potential security gaps before they are exploited by malicious actors.

In small critical infrastructure deployments such as microgrids, resource and time limitations often require the prioritization of vulnerability assessments and penetration testing. These efforts are vital because they indirectly assess the effectiveness of security controls. Penetration testing efforts focused on industrial control systems pose technical challenges because they demand deep knowledge and practical experience. The challenges are acerbated by the integration of information technology and operational technology that increase attack surface complexity and demand deep expertise and strong experience in both domains.

Systematic penetration testing engages a structured methodology involving predefined procedures and phases such as planning, reconnaissance and exploita-

tion to ensure that the testing is repeatable and consistent. It includes a process for prioritizing testing based on potential impacts, and it requires clear justifications for selecting scenarios, techniques and tools. The approach also emphasizes the thorough documentation of testing procedures, findings and recommendations. This structured methodology enables experienced and less experienced testers to conduct comprehensive assessments and articulate the rationale behind their test selections. A well-regarded exemplar is the Penetration Testing Execution Standard (PTES) [16].

As discussed in the next section, relatively few studies discuss systematic penetration testing of cyber-physical systems. However, they are primarily targeted at Internet-of-Things systems, not industrial control systems. The two types of systems may have similar attack surfaces, but they differ in complexity and requirements. Industrial control systems use proprietary protocols and have safety and availability requirements that render them more complex than Internet-of-Things systems. Additionally, performing industrial control system penetration testing requires considerable domain and tool expertise to avoid negative impacts such as critical service disruption and equipment damage. Furthermore, Internet-of-Things methodologies do not consider the valuable threat modeling insights gained during the design phase of industrial control systems.

A recent study [9] proposed a cyber-physical system threat modeling methodology that is applicable during the early design phase. Since detailed system information was not available, a high-level threat modeling method similar to STRIDE [6] was used. STRIDE covers six fundamental cyber security threats – spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege. It attempts to ensure that a designed system has six fundamental security properties – authentication, integrity, non-repudiation, confidentiality, availability and authorization.

STRIDE proved its effectiveness in a recent power grid case study [9]. It provides a common language that is understood by cyber security and technical specialists involved in threat modeling. The choice of a less sophisticated threat modeling method similar to STRIDE during the design phase facilitated brainstorming during the threat modeling activities and the ensuing discussions about threat impacts. However, as the system transitioned to the implementation and testing phases, it became clear that a more detailed model was necessary to effectively plan attack scenarios, sequences and paths.

Despite the importance of connecting early design-phase threat modeling with later-stage testing, practical and systematic penetration testing methods for industrial control systems that leverage threat modeling insights are lacking. To address the gap, this study investigates how the outcomes of industrial control system threat modeling performed during the design phase can support penetration test planning.

This work extends an earlier study [9] by proposing a seven-stage methodology for systematic penetration testing of newly-deployed industrial control systems based on threat modeling conducted during the design phase. The methodology translates threats elicited using a high-level model similar to STRIDE to

a medium-level model (i.e., MITRE ATT&CK for Industrial Control Systems (MITRE ATT&CK for ICS) [19]). The mapping facilitates the creation of comprehensive and justifiable testing scenarios that enhance the quality of industrial control system penetration testing. The methodology is validated using a real-world microgrid case study. The case study offers valuable insights into the practical challenges and nuances encountered in real-world deployments, contributing to a better understanding of the practical considerations involved in industrial control system penetration testing.

2 Related Work

Planning industrial control system penetration testing is challenging due to system complexity and the diverse technologies employed. The National Electric Sector Cybersecurity Organization Resource (NESCOR) [18] has developed a guide for conducting penetration tests of electric utility systems. It divides penetration testing tasks into five stages – penetration test scoping, architecture review, target system setup, end-to-end penetration test analysis and result interpretation and reporting.

The NESCOR guide places penetration testing tasks in four categories depending on the skills and time required to perform the tasks. The server operating system penetration task that requires basic testing skills should be performed frequently; the task is typically done by an internal team. The server application penetration task requires moderate testing skills and should be performed frequently. The network communications penetration task requires a high level of expertise and is performed occasionally. The embedded device penetration task requires highly specialized expertise; it is time-consuming and is performed rarely.

Several studies have leveraged threat modeling to plan penetration testing, mostly for Internet-of-Things systems. For example, Salzillo et al. [17] have proposed a systematic Internet-of-Things penetration testing methodology comprising four stages – system modeling, threat modeling, planning and penetration testing. The methodology was applied to penetrate a home automation environment using the MQTT protocol. It employed graph-based multi-cloud application composition modeling to elicit system threats using a threat catalog and common attack pattern enumeration and classification (CAPEC) [10]. Details of attack patterns for potential system threats were constructed, following which an attack plan was created.

PatrIoT is a systematic Internet-of-Things penetration testing methodology [20]. It involves four stages – planning, threat modeling, exploitation and reporting, each with well-documented substages. PatrIoT decomposes an Internet-of-Things environment into seven attack surfaces – hardware, firmware, network, web, cloud, mobile and radio. The methodology compiles the OWASP IoT Top 10 [14] into 100 security weaknesses based on the Common Weakness Enumeration (CWE) [11]. PatrIoT has been applied to several Internet-of-Things devices, including drones, refrigerators and cameras [20].

As noted by Fujimoto et al. [5], industrial control systems and information technology systems share certain similarities with regard to penetration testing, such as the use of common operating systems (e.g., Windows and Linux) and web applications. However, industrial control system penetration testing requires additional domain-specific knowledge due to specialized protocols and standards that are not encountered in information technology environments. Moreover, while information technology penetration testing primarily focuses on cyber aspects, industrial control system penetration testing must also consider the physical impacts of cyber attacks.

The testing objectives also differ. Information technology system penetration testing emphasizes confidentiality breaches whereas industrial control system penetration testing prioritizes data integrity and process availability. These priorities should be reflected in the design of industrial control system penetration testing scenarios. Fujimoto and colleagues also present an industrial control system penetration testing workflow and list tools for executing attacks.

Defining the scope of penetration testing and planning the tests with guidance from threat modeling can support testing priorities. Although a few studies address systematic penetration testing of cyber-physical systems, they often ignore threat modeling performed during the design phase when planning penetration tests. Incorporating the early-stage outputs into the later stages is essential to increase the effectiveness and efficiency of secure development lifecycle efforts. Additionally, there is a notable lack of real-world case studies that document systematic penetration testing efforts for industrial control systems.

3 Proposed Methodology

The proposed methodology for penetration testing newly-deployed industrial control systems is a systematic and structured approach involving seven sequential stages: (i) threat modeling, (ii) data flow diagram updating, (iii) penetration test scoping, (iv) penetration test planning, (v) penetration testing, (vi) vulnerability ranking and (vii) reporting.

Stage 1: Threat Modeling. The first stage involves threat modeling conducted during the industrial control system design phase. Khalil et al. [9] have presented a nine-stage cyber-physical system threat modeling methodology, along with a detailed case study that covers industrial control system threat modeling. The threat modeling results of Khalil and colleagues are incorporated in the scoping and penetration test planning stages of this research to facilitate targeted and effective penetration testing that contributed to a secure industrial control environment.

Stage 2: Data Flow Diagram Updating. The second stage involving data flow diagram updating begins after the industrial control system has been implemented. During the stage, the cyber security team collaborates with relevant

stakeholders to revise data flow diagrams and assess implementation decisions that may have affected the initial documentation and threat list. The stage also provides an opportunity to determine whether penetration testing should be conducted internally by the cyber security team or outsourced to a third party. Ideally, this decision should be made during the system implementation phase to allow adequate time for the stakeholders to engage and coordinate with external testers, if needed.

Stage 3: Penetration Test Scoping. The third stage involves defining the penetration testing scope in collaboration with the stakeholders. This includes identifying the targeted attack vectors such as remote access, Internet-facing applications and potential lateral movement paths. The type of penetration testing – white box, gray box or black box testing – should also be determined.

The principal candidates are white box and gray box testing. White box penetration testing, in which testers are granted access to system diagrams, configurations and code, is recommended. It is less time-consuming when an internal team does the testing, especially if the team was involved in the earlier threat modeling stage. Gray box testing is more appropriate when engaging third-party testers. In this case, testers receive limited information such as high-level design and critical information flows. It enables testers to focus on the key attack vectors without requiring full system disclosure. While white box testing provides comprehensive insights, the involvement of external testers raises issues such as security concerns, time required to gain familiarity with the systems and higher costs for extended engagements.

During the scoping stage, the stakeholders may also determine the complexity of the penetration testing tasks [18]. The decisions are often based on the system components and attack surfaces. This work adopts the categorization proposed by Searle et al. [18], which identifies four attack surfaces for electrical utilities and specifies the following four penetration testing tasks:

- **Server Operating System Penetration:** This task involves vulnerability assessments of Windows and Linux systems, especially identifying exploitable vulnerabilities that may provide system access. As noted by Searle et al. [18], the task typically involves information gathering, vulnerability analysis and vulnerability exploitation. Penetration testers are quite familiar with the task and several tools such as Metasploit are available to facilitate task execution.
- **Server Application Penetration:** This task involves testing server applications and identifying vulnerabilities via application mapping, application discovery and application exploitation [18]. The task is more complex and time-consuming than server operating system penetration.
- **Network Communications Penetration:** This task involves exploiting vulnerabilities in network protocols and communications media. The task is complex because it requires substantial expertise in network protocols and wireless attack tools given the reliance of industrial control systems on diverse wireless communications technologies, including Wi-Fi, cellular and Global Position System (GPS). Protocols such as Zigbee, IEEE 802.15.4, Modbus

and DNP3 are also widely used in industrial control systems and attacks on them require expertise related to their protocol datagrams and functionality.
- **Embedded Device Penetration:** This task is the most challenging and time-consuming of the four penetration testing tasks. It involves penetration tests of embedded and microprocessor-based devices, including firmware, input interfaces (serial and parallel), and storage media such as RAM, EEPROM and flash memory [18].

Decisions pertaining to penetration testing tasks involved in the scoping stage, establish expectations regarding the testing timeframe, required expertise, budget and expected outcomes. Omitting any of the four tasks means accepting the potential risks of undiscovered vulnerabilities in the associated areas. Any decision would, of course, require stakeholder approval.

Another key decision during the scoping stage pertains to the threats included in the testing plan. Industrial control system threats are documented during the threat modeling stage and data flow diagrams are updated during the subsequent stage. However, the final list of elicited threats can be extensive and vary in their severity. Therefore, selecting a prioritized subset of threats based on established criteria (e.g., criticality and impact) is essential to articulating a focused and efficient penetration testing effort. The selection process enables penetration testers to focus on the most significant threats while optimizing the testing process and maximizing its effectiveness.

Stage 4: Penetration Test Planning. The fourth stage involves penetration test planning based on the identified scope. Penetration testers typically rely on available guides and supporting frameworks. However, some guides, such as the NESCOR guide [18], provide high-level information without detailed procedures. STRIDE threat trees [6] also provide high-level guidance and incorporate testing criteria, but the threat trees primarily focus on software security. Detailed guides are also available – an example is the OWASP testing guide [15] that provides web, mobile application and firmware security testing details.

In practice, penetration testers often do not execute test planning in a structured or well-documented manner. One of the main contributions of this study is the demonstration of how an industrial control system testing plan can be guided by STRIDE threats categorized according to the MITRE ATT&CK for ICS framework [1]. Depending on the timeframe and agreements, penetration testers may also prepare a document outlining the testing plan, including the testing scenarios and their procedures. However, when penetration testing is performed by a third party, the outcome of the stage may be limited to a high-level plan outlining the types of attacks to be tested.

Stage 5: Penetration Testing. During the fifth stage, penetration testing, the tester performs the planned tests to identify vulnerabilities in the industrial control system environment and documents the findings. Based on the findings, the testing plan may be updated to address newly-discovered attack vectors.

Stage 6: Vulnerability Ranking. The sixth stage involves the analysis and prioritization of the discovered vulnerabilities based on their potential impacts. This ranking may be conducted using the Common Vulnerability Scoring System (CVSS) [3] or a custom method that considers the specific impacts to the industrial control system environment. This prioritization ensures that mitigation efforts are focused on the most critical and highly-ranked vulnerabilities.

Stage 7: Reporting. The final seventh state involves the documentation of all the findings in a comprehensive report for the stakeholders. This report typically includes detailed descriptions of identified vulnerabilities, recommended remediation actions and risks associated with the discovered vulnerabilities. The three stages – penetration testing, vulnerability ranking and reporting – can be executed sequentially or in parallel depending on project complexity and resource availability.

4 Microgrid Case Study

This section describes a penetration testing case study involving an operational microgrid. The penetration testing process was a collaborative effort involving four stakeholder groups: (i) a cyber security team that included the first and second authors of this chapter, (ii) an electric power systems expert (third author) who represented the power engineering team and the asset owner, (iii) a software architect and (iv) a third-party penetration tester.

The penetration testing was conducted over a period of eight months, from February 2023 to September 2023. During this period, the stakeholders participated in 13 technical meetings, during which the details of each stage were discussed. The meetings focused on technical aspects and did not include managerial discussions related to the selection of the testing entity and contractual agreements.

The first stage involving threat modeling during the design phase is discussed in detail in previous work [9]. Therefore, this case study starts with data flow diagram updating, the second stage of the methodology.

4.1 Data Flow Diagram Updating

During this stage, the cyber security team collaborated with the software architect and electric power systems expert to discuss the design changes implemented since the initial data flow diagram creation. The updates, conducted late in the implementation phase, sought to track the changes made during the implementation that affected the cyber security documents and data flow diagrams. This ensured that the data flow diagrams accurately reflected the system state before initiating efforts in the subsequent penetration test scoping and planning stages.

Based on the resource and time constraints, the stakeholders prioritized penetration testing using the IEC 62443 security verification and validation testing processes. This meant that security requirements and threat mitigation tests

were excluded from the project. Negotiations to determine whether the penetration testing would be conducted internally by the cyber security team or outsourced to a third party were concluded during system implementation.

The stakeholders chose to involve a third-party penetration tester to provide a fresh perspective regarding the system security posture. Additionally, an individual with specialized penetration testing skills and experience could be effective given the limited timeframe assigned to testing. This solution would ensure better resource management, especially considering the limited number of internal cyber security experts involved in the effort.

Negotiations with third-party penetration testers proceeded in parallel with data flow diagram updating. The final agreement was reached with a professional freelance penetration tester with experience primarily in information technology system penetration testing and an operational technology background. Once the agreements and data flow diagram updates were finalized, the penetration tester was provided with the updated data flow diagrams, data flows and prioritized threat list. Providing the threat modeling results at the beginning of the effort contributed to the tester's understanding of system threats and their potential impacts. Follow-up meetings were scheduled to discuss the provided documents in detail and address questions raised by the tester. The data flow diagram updating stage ensured that all relevant documents were updated and that the tester fully grasped the data flow diagrams and system threats.

4.2 Penetration Test Scoping

Stakeholder meetings were scheduled to discuss the penetration testing scope. The meetings, which included the penetration tester, focused on potential attack surfaces (e.g., Internet-facing web applications), attack vectors (e.g., unpatched vulnerabilities and weak credentials), attack agents (e.g., cyber criminal groups) and attack scenarios. The efforts prioritized penetration testing of the operating system, server applications and network communications. Embedded devices penetration testing and physical attacks were excluded from the testing scope.

It was determined that gray box testing would be conducted. Data flow diagrams, network diagrams and threat lists were provided to the tester, and questions posed by the tester were addressed. However, source code was not provided to the tester. Discussions also centered on the testing location, ultimately deciding on remote testing and excluding physical on-site testing. Social engineering attacks (e.g., phishing) were excluded because they rely on user interaction that was outside the scope of the project because the microgrid system was not yet operational.

The scoping discussions also focused on critical threats. The specific data flow diagram elements that would be within scope were negotiated and their threat consequences were discussed. Attack scenarios based on the threats were articulated along with considerations about how the attacks might be reflected on the system side and how the penetration tester would ensure that the attacks were successful. Critical threats guiding the testing plan were emphasized while providing the tester with the flexibility to explore threats outside the provided

threat list. Additional discussions covered the testing timeframes, critical vulnerability reporting procedures and the protocol for handling denial-of-service attacks.

The stakeholders then discussed how the tester would gain initial access to the system and all the relevant attack vectors were considered. The tester's access (e.g., virtual private network and virtual machine) and permissions (e.g., regular user and admin) were determined. To enable the tester to observe the impacts of attacks on system alerts, regular user access was required as well as admin access to certain tools.

The discussions also addressed the depth of testing, considering whether to focus solely on initial device access or also include lateral movements. It was decided to provide the tester with limited access accounts that would allow initial access to some devices. This approach would enable the tester to assess the possibility of lateral movements within the system.

In summary, the penetration test scoping stage defined the testing type, boundaries (i.e., out-of-scope activities), critical attacks to be executed and their consequences, rules of engagement, emergency contacts and expectations about the penetration testing timeframe.

4.3 Penetration Test Planning

The cyber security team was keen on structuring the penetration test planning effort based on an attack database that complemented the penetration tester's knowledge. However, the selection of a suitable attack database was a major challenge. During the design phase, the microgrid attack taxonomy proposed in previous work [2] was updated. The high-level taxonomy proved valuable for threat elicitation in the threat modeling case study [9]. However, it lacked details about attack execution that are crucial to penetration test planning. For this reason, the use of STRIDE threat trees [6] as the attack database were investigated. The trees were discussed during two meetings with the tester. While the trees provided a structured approach, their software-centric orientation overlooked several industrial control system attack scenarios. This limitation would have resulted in an incomplete and potentially ineffective penetration testing plan. Consequently, the STRIDE trees were excluded from the attack database.

STRIDE is a high-level threat analysis model [19] that is suited to threat elicitation during the design stage, when detailed system specifications (e.g., vendors, operating systems and protocols) are unavailable. However, this information is available after system implementation and during penetration test planning. Recognizing the limitations of the high-level microgrid attack taxonomy used in the threat modeling case study [9] and STRIDE threat trees, alternative approaches were explored to create a lower level model with rich attack execution details.

Because the tests were to be performed by a third party, the cyber security team decided to formulate the testing plan without imposing specific scenarios and tools as these are typically the responsibility of the tester. For this reason, it was decided to use MITRE ATT&CK for ICS [19], a mid-level modeling framework that provides details about attack techniques. However, MITRE

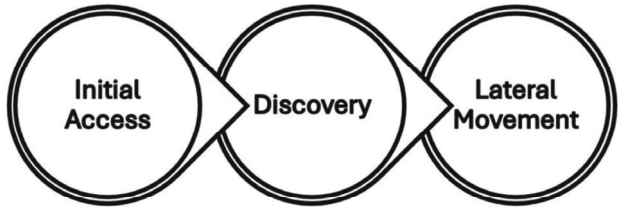


Fig. 1. Attack precondition stages

ATT&CK for ICS is not designed specifically for penetration testing and lacks detailed information about test execution. While it identifies attack techniques, the penetration tester would have to identify the appropriate tools to conduct the various exploits.

Since the threat elicitation leveraged STRIDE, mapping the identified threats to MITRE ATT&CK for ICS tactics and techniques became crucial. At that stage, it was decided to have focused meetings between the cyber security team and the penetration tester during which MITRE ATT&CK for ICS would be investigated in detail and mapped to the STRIDE threats. These efforts guided the penetration tester in planning the tests in a structured manner.

MITRE ATT&CK for ICS was mapped using MITRE ATT&CK Navigator [13] by going through each technique description and adding a comment about the technique categorization (spoofing, tampering, repudiation, information disclosure, denial of service or elevation of privileges). Some techniques were assigned to more than one STRIDE category (e.g., the adversary-in-the-middle technique was classified as a technique that could be used in spoofing, tampering, information disclosure and denial of service).

The cyber security team and the penetration tester discussed the proposed categorization and agreed on the techniques to be included in the test scope. For each technique, a decision was made to include or exclude it from the penetration tests. Some techniques were determined to require more time and effort than others and required certain preconditions to be met in order to be successful. For example, operational technology attack tests require initial access to the operational technology network without which the penetration tester would be unable to apply the appropriate techniques. Such tests were marked for potential inclusion in the penetration testing plan if the time constraints and other required preconditions were met. Attack preconditions techniques were also added to the categorization.

The following subsections present the STRIDE categorization of MITRE ATT&CK for ICS techniques and their preconditions.

Attack Preconditions. As discussed in the threat modeling case study [9], except for certain types of denial-of-service attacks, the attacker has to first gain access to a system to cause harm; this is considered to be an attack precondition. Figure 1 shows the three attack precondition stages – initial access, discovery and

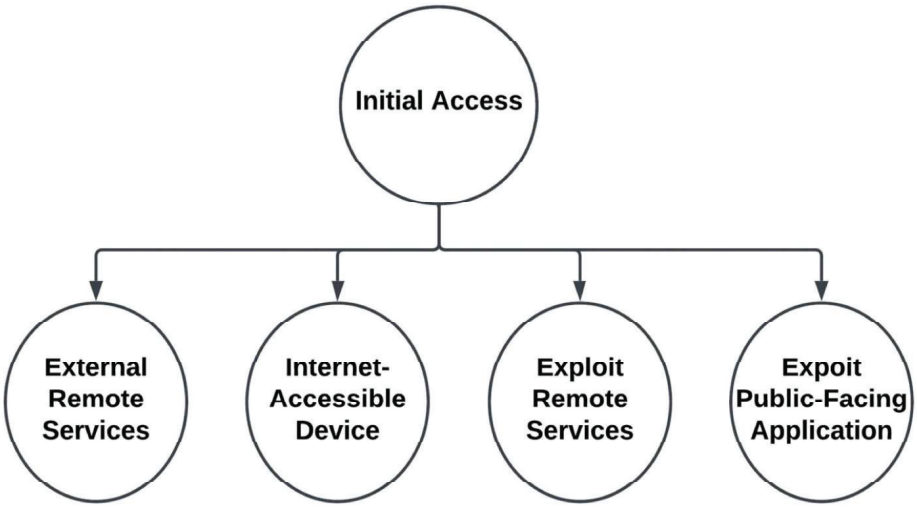


Fig. 2. MITRE ATT&CK for ICS initial access techniques in the testing plan.

lateral movement. These stages are also defined as tactics in MITRE ATT&CK for ICS.

In a possible attack scenario, initial access to the system would be gained using an initial access technique such as exploiting a public-facing application over the Internet. Next, system discovery would be performed using network sniffing. Following this, a lateral movement would be executed by leveraging default credentials or sniffed valid accounts.

Aside from the STRIDE categorization of MITRE ATT&CK for ICS, three trees were created for the possible precondition techniques that could be used by the tester to gain system access. The three trees, corresponding to initial access, discovery and lateral movement, included selected techniques from MITRE ATT&CK for ICS. Only the techniques selected for the penetration testing are described here. The complete lists of techniques for the three chosen tactics are provided in the MITRE ATT&CK for ICS matrix [12].

It was decided that the penetration tester would gain initial access by simulating an attack on the system over the Internet. Figure 2 shows the relevant techniques related to the MITRE ATT&CK for ICS initial access tactic that were selected. The external remote services technique targets services such as virtual private networks potentially using valid accounts. The Internet-accessible device technique seeks to gain access to Internet-exposed devices, intentionally or intentionally, such as cellular modems without using exploits. The remaining two techniques, exploit remote services and exploit public-facing application, leverage software vulnerabilities to gain unauthorized access to the system.

The external remote services technique was eliminated from the tree because it targets internal services typically used by operators for remote management, which renders it more relevant to lateral movement in the context of this study. Additionally, the MITRE ATT&CK for ICS initial access techniques that require physical presence (e.g., rogue master), user interaction (e.g., spearphish-

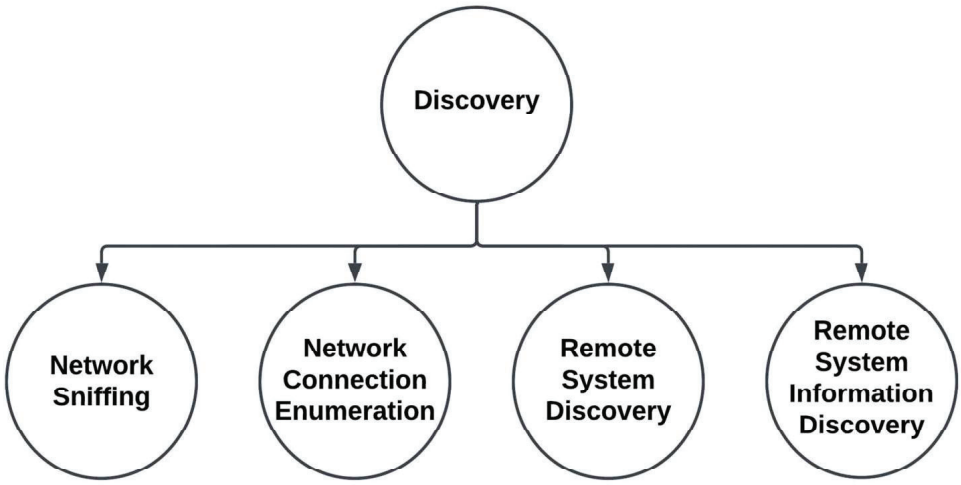


Fig. 3. MITRE ATT&CK for ICS discovery techniques in the testing plan.

ing attachment, drive-by compromise) and outside the testing scope (e.g., supply chain compromise) were also excluded from the tree.

If the tester were to gain remote access to the system from the Internet, network discovery should then be used to identify additional vulnerabilities that could be exploited for lateral movement. However, if all the initial access attacks were to be unsuccessful, the tester could use the provided limited access account to establish initial access and proceed with network discovery.

Figure 3 shows the relevant techniques selected from the MITRE ATT&CK for ICS discovery tactic. Network sniffing is a fundamental technique for collecting information about the system. Network connection enumeration may be used by the penetration tester to inspect the state of network connections and collect information about device roles. The technique might also be used to collect information about the subnets and internal domains. The remote system discovery and the remote system information discovery techniques may be employed to discover the remote services and collect detailed information about them (e.g., configurations).

Following the discovery stage, the tester might attempt to pivot between devices on the same network using lateral movement techniques. It was also agreed that the testing scope should encompass lateral movement attempts across distinct network segments. Figure 4 shows the four lateral movement techniques selected in the testing plan along with two additional techniques (light gray) identified as potential options for further testing.

Information gathered during the discovery of devices and operating systems can help identify default passwords, enabling the tester to employ the default credentials technique for lateral movement. Similarly, the enumerated user or service accounts during discovery can be leveraged in the valid accounts technique. The exploit remote services technique, previously used for initial access,

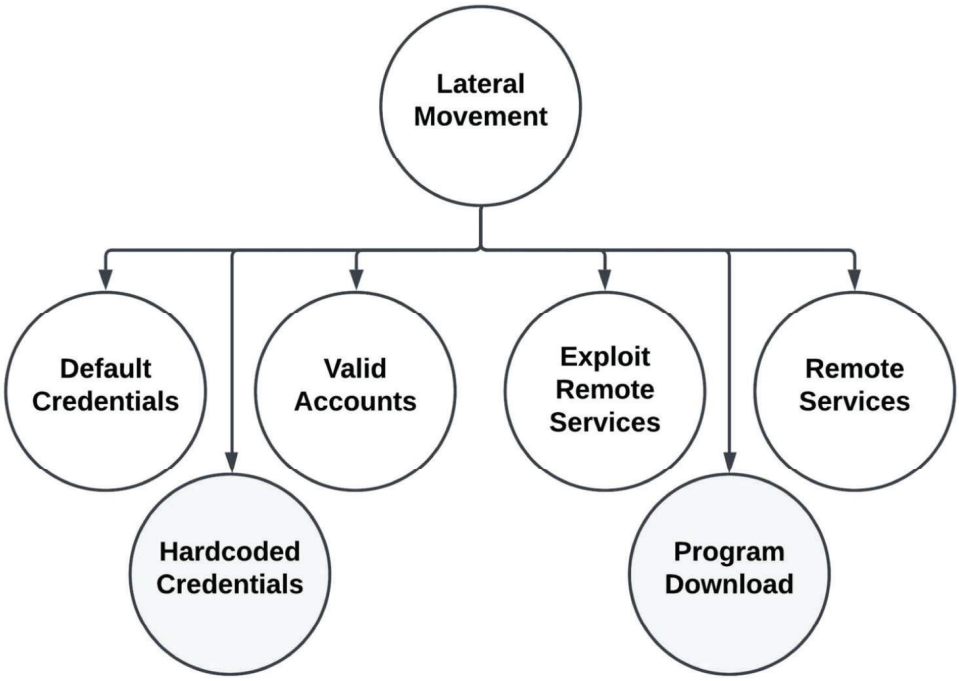


Fig. 4. MITRE ATT&CK for ICS lateral movement techniques in the testing plan.

is applicable for lateral movement within the system. Additionally, the remote services technique may be used to navigate the network laterally.

The hardcoded credentials technique was marked as a potential test depending on the available timeframe (light gray). Exploiting the credentials might involve leveraging vulnerabilities to reveal them, which would require advanced expertise. The program download technique was identified as a potential test due to its relevance in industrial control system environments (light gray). The technique has the precondition of initial access to the operational technology network.

STRIDE Tests. To facilitate the alignment between the threats identified during the threat modeling phase and the penetration testing plan, the various MITRE ATT&CK for ICS techniques were mapped to STRIDE categories. This enabled the prioritization and the evaluation of the relevance of each categorized technique in the testing plan. However, this introduced a challenge because not all the techniques could be mapped directly to STRIDE. For example, some techniques under the execution tactic, such as those involving command-line and graphical user interfaces, describe how an unauthorized actor (penetration tester in this case study) would interact with the system. While these techniques may serve as supporting attributes in the STRIDE categories, they do not align directly with the core STRIDE threat types.

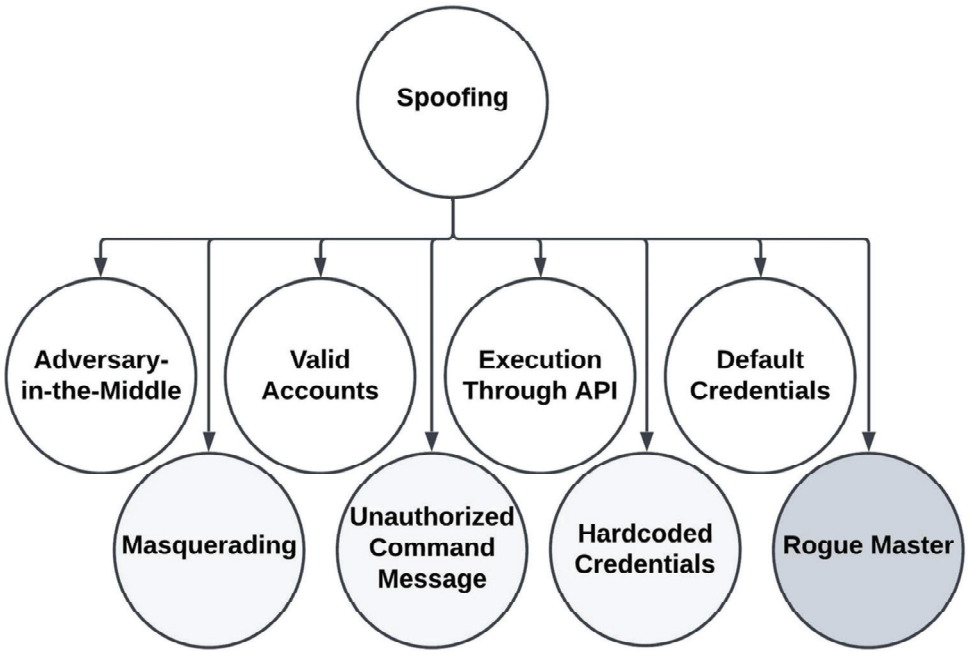


Fig. 5. Categorization of spoofing techniques.

The figures in this section present the STRIDE categorizations of the selected MITRE ATT&CK for ICS techniques. The figures are color coded to differentiate between techniques included in and excluded from the testing plan. Techniques selected for testing are unmarked whereas those outside the scope of the case study scope are marked in dark gray. Techniques with light gray markings are potentially included in the testing (i.e., optional) if the associated preconditions are met. These preconditions might involve factors such as additional testing time or achieving specific initial access to a device in the network.

Note that some techniques were repeated in the STRIDE categorizations. For example, the rogue master technique was categorized as spoofing and tampering, depending on how it would be used. Also, the adversary-in-the-middle technique plays a key role in testing various attack vectors. Therefore, it was incorporated in the spoofing, tampering, information disclosure and denial-of-service categorizations of the MITRE ATT&CK for ICS techniques.

Figure 5 shows the eight techniques categorized as spoofing techniques. The spoofing testing plan included four core techniques and three potential techniques. Successful use of the adversary-in-the-middle technique could also serve as a proof of concept for successful masquerading. Therefore, masquerading was marked as a potential test to be performed if time permitted. Prioritizing the testing of the in-house control software led to the prioritization of relevant techniques similar to valid accounts and execution through the application programming interface (API). The unauthorized command message technique was marked for potential testing because it requires initial access to the operational technology network that would not be provided to the tester. Finally, the rogue

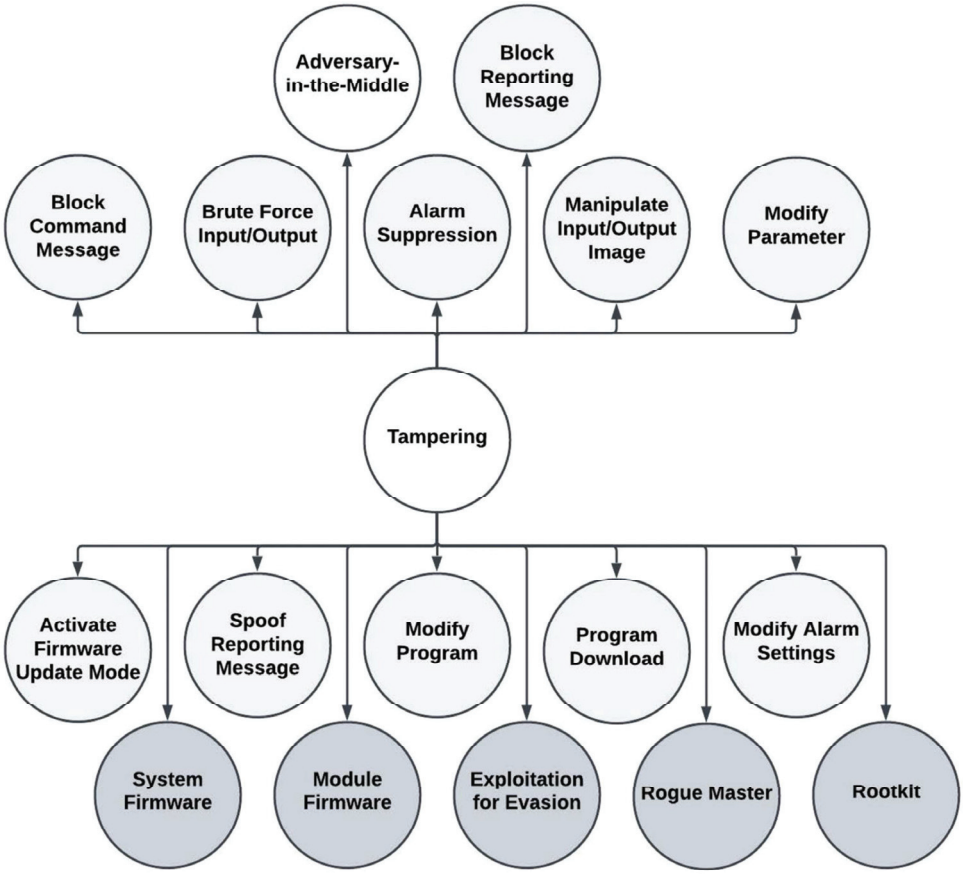


Fig. 6. Categorization of tampering techniques.

master technique was excluded from the testing scope because the plan only involved simulating remote attacks absent physical access to the tester.

Spoofing threats were the most critical threats identified in the threat modeling stage. The tester could simulate attacker behaviors using the spoofing categorization of MITRE ATT&CK for ICS techniques. For instance, the tester could combine the adversary-in-the-middle technique with the masquerading technique to mimic a threat agent who gains unauthorized access and potentially disrupts critical operations by masquerading as a microgrid administrator. Additionally, the tester could evaluate the security of API endpoints by attempting to execute commands and assessing whether authentication is required. Techniques that use default device passwords (i.e., default credentials) or brute-forcing accounts (i.e., valid accounts) could also be used to identify potentially weak authentication mechanisms.

Figure 6 shows the 17 techniques categorized as tampering. However, only the adversary-in-the-middle technique was explicitly included in the tampering testing plan. The categorization of tampering techniques sparked interesting dis-

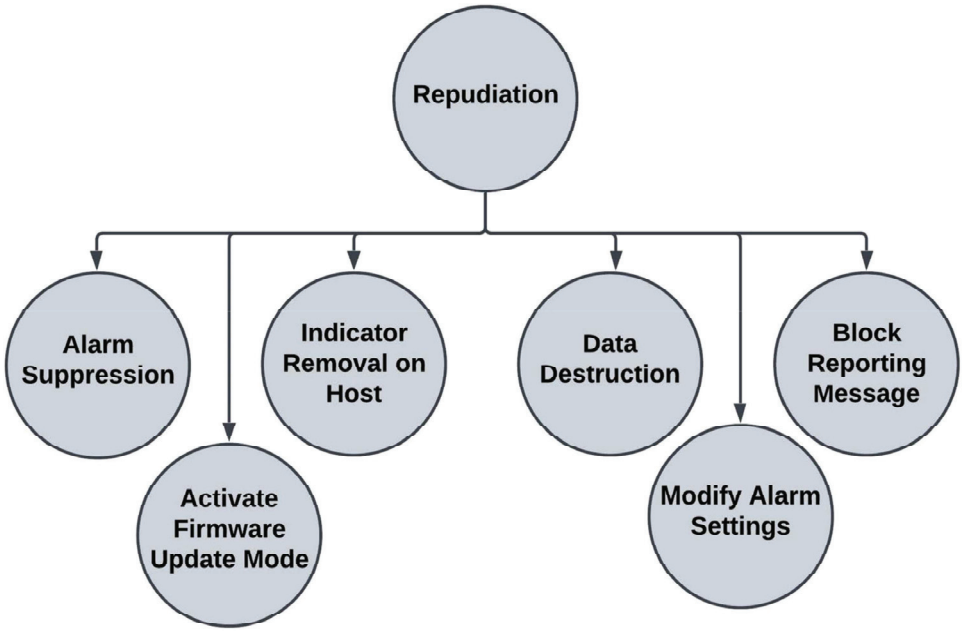


Fig. 7. Categorization of repudiation techniques.

cussions. Techniques similar to block reporting message might seem more suited for denial-of-service attacks because they disrupt monitoring services. However, it was decided to also include the technique in the tampering category. Consider a scenario where a system configuration always displays the last received monitoring data without raising alarms for missing reports. In such a case, blocking reporting messages during an attack could be considered tampering with the ability of the system to report accurate information. The same logic applies to the block command message technique. For example, if an attacker were to block a message containing a new setpoint value (a desired operating condition), the system would not receive the update and would remain in the old configuration. This effectively tampers with the system configuration established by the administrator.

Other tampering techniques were de-prioritized because it was decided that performing the adversary-in-the-middle technique could be used as a proof of concept for techniques similar to block reporting message and block command message in the event that the testing timeframe did not enable the tests to be conducted. The system firmware and module firmware techniques were eliminated because the embedded device (including firmware) attack surface was out of the scope of the testing plan.

Figure 7 shows the six techniques categorized as repudiation techniques. However, it was decided to exclude all the techniques related to repudiation from the testing plan because the losses and consequences of the repudiation threats were minimal compared with the other threats. The repudiation techniques mainly

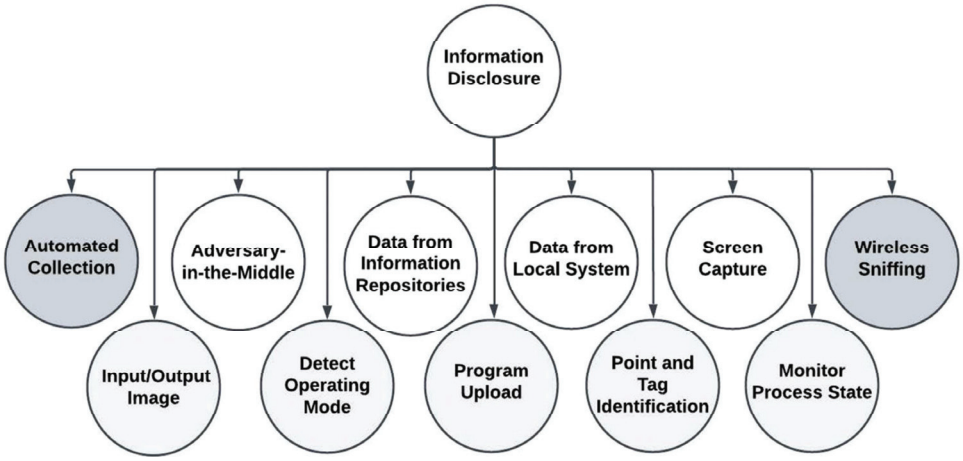


Fig. 8. Categorization of information disclosure techniques.

sought to hide attacker traces. This was out of the scope of the testing plan because the main goal was to simulate attacks that might cause critical system impacts.

Figure 8 shows the 11 techniques categorized as information disclosure techniques. During the threat modeling stage, information disclosure threats were classified as low priority due to their limited impact on system functionality. However, four information disclosure techniques were still included in the testing plan – adversary-in-the-middle, data from information repositories, data from local system and screen capture. In real-world attack scenarios, the screen capture technique is often combined with malware infection or other automation techniques; it is vital in penetration testing as well. The primary reason for incorporating the four techniques was to understand the potential information an attacker could access and how the collected information could be leveraged attack progression.

However, five information disclosure techniques were de-prioritized – input/output image, detect operation mode, program upload, point and tag identification, and monitor process state. The application of these techniques requires prior access to the operational technology network, but this precondition cannot be taken for granted because the tester may not necessarily be able to establish such access. Âă The automated collection technique was excluded due to the time commitment required for scripting and automation tool development, and due to the low priority of information disclosure threats.

Figure 9 shows the nine techniques categorized as denial-of-service techniques. Since the target system was still under development during the testing, it was decided to include some of the techniques in the plan to assess the system. Again, the success of the adversary-in-the-middle technique was considered to be a proof of concept for some other techniques. For example, if the tester were to spoof the default gateway MAC address with a device that cannot process

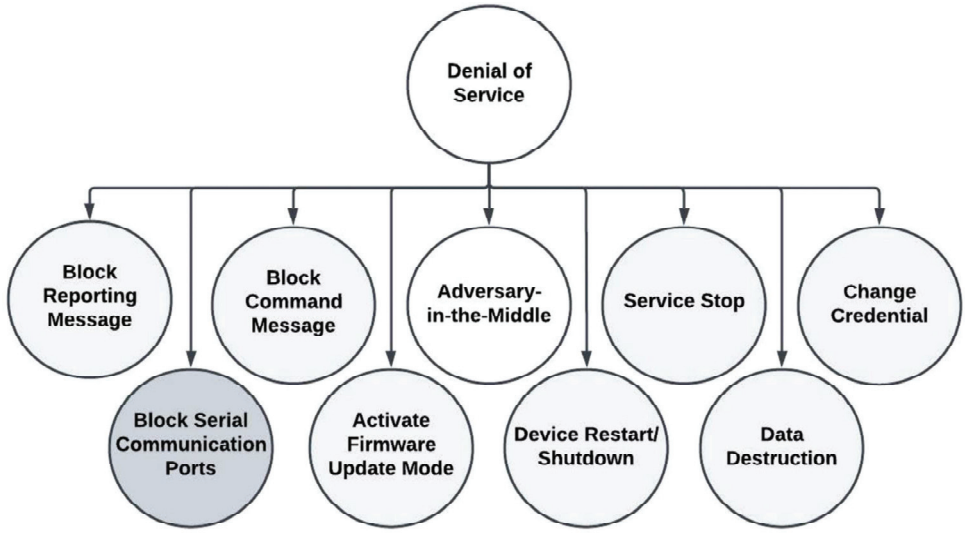


Fig. 9. Categorization of denial-of-service techniques.

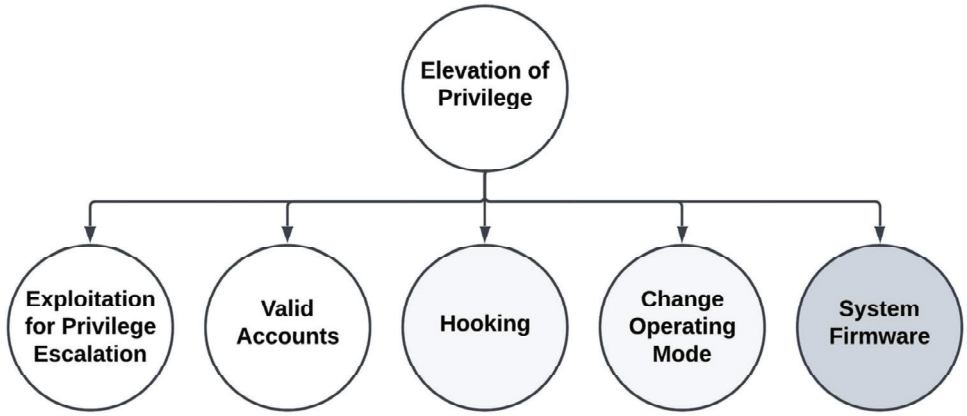


Fig. 10. Categorization of elevation of privilege techniques.

traffic, a denial-of-service condition would result. The use of the adversary-in-the-middle technique in such a case proves that other techniques similar to block reporting message and block command message are also possible.

Figure 10 shows the five techniques categorized as elevation of privilege techniques. The exploitation for privilege escalation and valid accounts techniques were added to the testing plan. The hooking and change of operation mode techniques were potential additions in case the tester gained access to the operational technology network.

At the end of the penetration testing period, the cyber security team discussed the categorized techniques again with the tester. The techniques were then categorized with a new color scheme with different colors used to distin-

guish between successful attacks, failed attacks, attacks excluded due to time limitations and attacks not performed because their preconditions were not met. This was done to obtain better visualizations of the testing outcomes and support planning for future exercises. It could also focus attention on critical tests that failed or were not included in the penetration testing plan.

4.4 Vulnerability Ranking

Twelve software vulnerabilities were reported by the penetration tester. It was interesting that all the reported vulnerabilities were not direct operational technology vulnerabilities albeit they could have severe impacts on system safety and availability. The reported vulnerabilities were categorized by the tester using the CVSS v3.1 calculator [4] under the base metric.

4.5 Reporting

The penetration tester provided a report documenting the testing methodology, the scope of the tests and the assumptions made. The report also included details about the discovered vulnerabilities, including the proofs of concept and CVSS v3.1 base scores, and the recommendations for mitigation. Following the report submission, two meetings were held with the penetration tester to review the discovered vulnerabilities and discuss their impacts. A separate meeting was held to document the tests executed during penetration testing, based on the STRIDE categorizations of the MITRE ATT&CK for ICS techniques. The meetings discussed the specific techniques that were used and the techniques excluded due to time constraints or unmet preconditions. This clarified the scope of the testing performed and would also be valuable for planning future penetration tests.

The penetration testing report identified six critical and high-severity vulnerabilities. A separate review was conducted to examine the vulnerabilities and their mitigation strategies. The review compared the proposed mitigations for each reported vulnerability against the security requirements established during the design phase. The goals were to determine if the proposed mitigations were already addressed by the security requirements and to highlight potential gaps in the requirements. Additionally, the review assessed whether the security requirements were overlooked or were not implemented correctly.

The review revealed that mitigation techniques for 10 of the 12 reported vulnerabilities were already established during the requirements phase discussed in [9]. For the remaining two low-severity vulnerabilities, the system security requirements offered partial mitigation, but further action would have still been taken by the asset owner. Nevertheless, the review could be considered to validate the effectiveness of the established security requirements. This emphasizes the value of penetration testing in identifying overlooked or insufficiently addressed requirements. It also demonstrates how a structured penetrating testing approach can uncover critical issues and provide feedback to enhance future system design and requirements specifications.

5 Discussion

Threat modeling is a theoretical process that seeks to ensure the mitigation of critical threats during system design. However, the threat modeling process alone cannot guarantee that a deployed system will have the minimum acceptable level of security. Despite providing the security requirements to the design team as an outcome of the threat modeling exercise, some of the requirements could have been skipped later due to budget or time restrictions or they could have been applied incorrectly during system design. Therefore, pre-operational system security testing is crucial to ensure the minimum acceptable security level.

The IEC 62443 standard describes four testing processes for security verification and validation testing – security requirements, threat mitigation, vulnerability discovery and penetration testing. However, in small industrial control system projects with limited resources (similar to the current case study), stakeholders might combine the processes to streamline security assessments. Given the resource constraints, this case study prioritized penetration testing over security requirements and threat mitigation testing. The results demonstrate that penetration testing could reveal information related to unimplemented security requirements and poorly implemented mechanisms for mitigating critical threats.

When possible, security requirements and threat mitigation testing processes should be executed before penetration testing. Performing these two processes early can reveal critical vulnerabilities that were meant to be mitigated by design, enabling system owners to address them before penetration testing. This would enable penetration testers to perform more advanced attacks, potentially discovering critical attack vectors not considered or prioritized lower during threat modeling. Such vulnerabilities could result from unforeseen system interactions, configuration weaknesses and zero-days. In such cases, the penetration testing results can further validate or refine the threat priorities identified during threat modeling. If a seemingly low-priority threat is shown to have a critical impact during testing, it can be addressed with greater urgency.

Due to the inherent complexities of a real-world case study, the research encountered challenges that redirected the investigation along unforeseen directions. While several threat modeling methods exist for cyber-physical systems, STRIDE was selected as an easy-to-adapt approach for the threat modeling phase. STRIDE focuses on basic cyber security concepts that are quickly assimilated by stakeholders with limited cyber security expertise. Notably, during the early design phase, a high-level threat modeling method was required to provide security requirements. A more detailed taxonomy could have elicited a broader range of threats, which might have increased the complexity and time commitment of the threat modeling exercise, potentially hindering its use in future projects. In contrast, the efforts showed that penetration testing planning requires a "middle-level" taxonomy that provides details about industrial control system attack types while remaining manageable for high-level plan creation, especially because the testing was conducted by a third party. These complications justify the mapping of MITRE ATT&CK for ICS techniques to STRIDE threats.

Engaging a third party for penetration testing brings valuable expertise to bear, but it has some limitations. While the cyber security team might strive to guide the testing plan to cover information technology and operational technology security, a key challenge is to strike the right balance between the information technology and operational technology testing efforts. Maintaining a high-level of involvement during the planning stage is recommended for effective collaboration with external testers. However, it limits the ability of the cyber security team to create a detailed testing plan. High-level planning with the tester would ensure that the critical threats were tested, but the tester may still prioritize certain types of tests over others. Overlooking operational technology tests could potentially result in a bias towards information technology vulnerabilities compared with operational technology vulnerabilities. However, the case study demonstrates that information technology vulnerabilities could still have a high impact on industrial control system security.

The inherent sensitivity of industrial control system penetration testing reports poses challenges to academic publication. This chapter has proposed a systematic penetration testing methodology for newly-deployed industrial control systems built on threat modeling outcomes during the design phase. Penetration testing procedures have been discussed without disclosing system details such as network protocols, vendors and device models. System components targeted by the tests (successful or not) and details about the discovered vulnerabilities have not been provided. Instead, the STRIDE categorization of MITRE ATT&CK for ICS techniques has been elucidated to demonstrate that it can contribute to effective and efficient penetration testing. This deliberate approach protects the system from targeted attacks while providing insights into all the aspects of penetration testing.

6 Conclusions

Planning penetration testing for complex industrial control systems environments presents several challenges, especially when relying solely on collaborative plan creation with third-party testers. The seven-stage methodology for penetration testing newly-deployed industrial control systems leverages threat modeling outcomes from the design phase to define the penetration testing scope and guide test choices, leading to an efficient testing process. The methodology, which relies on a novel MITRE ATT&CK for ICS to STRIDE mapping benefits security practitioners by enhancing planning efficiency and streamlining collaboration with third-party penetration testers. The practical application of the methodology on a newly-designed microgrid system demonstrates its ability to help deploy detailed, effective and efficient industrial control system penetration testing efforts.

Future work will focus on developing a comprehensive library of penetration testing procedures tailored to industrial control system environments. By improving the planning process and potentially reducing the risk of missed vulnerabilities, the research has the potential to enhance the security postures of industrial control systems.

Acknowledgements. The authors acknowledge Alvaro Wim Schuller Fernandez for conducting the penetration tests and for his valuable contributions during the penetration testing efforts. This work was supported by the Estonian Ministry of Education and Research and the European Regional Development Fund under Grant no. 014–2020.4.01.20–0289 and by the European Union under Grant no. 2021-2027.6.01.23-0034.

References

1. Alexander, O., Belisle, M., Steele, J.: MITRE ATT&CK for Industrial Control Systems: Design and Philosophy, MITRE Product MP01055863. MITRE, McLean, Virginia (2020)
2. Bahsi, H., Ochieng'Dola, H., Khalil, S., Korotko, T.: A cyber attack taxonomy for microgrid systems. In: Proceedings of the Seventeenth Annual System of Systems Engineering Conference, pp. 324–331 (2022)
3. Forum of Incident Response and Security Teams (FIRST), CVSS: Common Vulnerability Scoring System v3.1, Specification Document – Revision 1, Cary, North Carolina (2019). www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf
4. Forum of Incident Response and Security Teams (FIRST), Common Vulnerability Scoring System Version 3.1 Calculator, Cary, North Carolina (2025). www.first.org/cvss/calculator/3.1
5. Fujimoto, M., Itani, Y., Mitsunaga, T.: Lessons learned for practical penetration test against industrial control systems. In: Proceedings of the IEEE International Conference on Computing, pp. 59–64 (2023)
6. Howard, M., Lipner, S.: The Security Development Lifecycle. Microsoft Press, Redmond, Washington (2006)
7. International Electrotechnical Commission: IEC 62443-4-1:2018: Security for Industrial Automation and Control Systems - Part 4–1: Secure Product Development Lifecycle Requirements. Switzerland, Geneva (2018)
8. Khalil, S., Bahsi, H., Korotko, T.: Threat modeling of industrial control systems: a systematic literature review. Comput. Secur. **136**, 103543 (2024)
9. Khalil, S., Bahsi, H., Ochieng'Dola, H., Korotko, T., McLaughlin, K., Kotkas, V.: Threat modeling of cyber-physical systems - a case study of a microgrid system. Comput. Secur. **124**, 102950 (2023)
10. MITRE, Common Attack Pattern Enumeration and Classification (CAPEC), Bedford, Massachusetts (2023). https://capec.mitre.org/index.html
11. MITRE, Common Weakness Enumeration (CWE), Bedford, Massachusetts (2024). https://cwe.mitre.org/
12. MITRE, MITRE ATT&CK Matrix for ICS, Bedford, Massachusetts (2025). https://attack.mitre.org/matrices/ics/
13. MITRE, MITRE ATT&CK Navigator, Bedford, Massachusetts (2025). https://mitre-attack.github.io/attack-navigator/
14. OWASP Foundation: OWASP Internet of Things Project, Wilmington, Delaware (2019).https://owasp.org/www-project-internet-of-things/
15. OWASP Foundation, WSTG – Latest, Wilmington, Delaware (2025). https://owasp.org/www-project-web-security-testing-guide/latest/
16. Penetration Testing Execution Standard Contributors, PTES Technical Guidelines (2012). www.pentest-standard.org/index.php/PTES_Technical_Guidelines

17. Salzillo, G., Rak, M., Moretta, F.: Threat modeling based penetration testing: the open energy monitor case study. In: Proceedings of the Thirteenth International Conference on Security of Information and Networks, article no. 8 (2020)
18. Searle, J., Rasche, G., Wright, A., Dinnage, S.: NESCOR Guide to Penetration Testing for Electric Utilities, Version 3, National Electric Sector Cybersecurity Organization Resource, Electric Power Research Institute, Palo Alto, California (2016). https://smartgrid.epri.com/doc/nescorguidetopenetrationtestingforelectricutilities-v3-final.pdf
19. Strom, B., Applebaum, A., Miller, D., Nickels, K., Pennington, A., Thomas, C.: Mitre ATT&CK: Design and Philosophy, MITRE Product MP180360R1. MITRE, McLean, Virginia (2020)
20. Suren, E., Heiding, F., Olegard, J., Lagerstrom, R.: PatrIoT: practical and agile threat research for IoT. Int. J. Inf. Secur. **22**, 213–233 (2023)

Appendix 5

V

S. M. Khalil, H. Bahsi, and T. Korõtko. Bridging industrial control systems design and testing through threat modeling-driven penetration testing – a microgrid case study. *Computers & Security*, 167:104906, 2026

Computers & Security 167 (2026) 104906

Contents lists available at ScienceDirect

Computers & Security

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

Full Length Article

Bridging industrial control systems design and testing through threat modeling-driven penetration testing - a microgrid case study

Shaymaa Mamdouh Khalil [a,*], Hayretdin Bahsi [a,b], Tarmo Korõtko [c]

[a] School of Information Technologies, Tallinn University of Technology, Estonia
[b] School of Informatics, Computing, and Cyber Systems, Northern Arizona University, United States
[c] School of Engineering, Tallinn University of Technology, Estonia

ARTICLE INFO

Keywords:
Threat modeling
Cyber-physical systems (CPS)
Industrial control systems (ICS)
Penetration testing
STRIDE
MITRE ATT&CK for ICS

ABSTRACT

Threat modeling is a widely recommended approach for supporting penetration test planning, while penetration testing can, in turn, verify and validate design-phase threat modeling outcomes. Yet, this interrelation remains largely overlooked in academic research. To address this gap, this study proposes BRIDGE, a nine-stage threat modeling-driven penetration testing methodology that connects the design and testing phases. Guided by design-phase threat modeling results, the methodology supports structured test selection, enabling a more focused and efficient testing process. The study introduces a mapping of MITRE ATT&CK for ICS techniques to STRIDE threat categories, establishing a key link between high-level threat analysis performed during system design and the intermediate-level attack representation required for penetration test planning. The methodology's practical applicability is demonstrated through a real-world case study of a recently deployed microgrid system. The study also examines the effectiveness of CVSS v4.0 compared to v3.1 in representing the distinctive risk profile of ICS vulnerabilities, considering both security requirements and potential safety impacts. This research provides practical guidance for ICS cybersecurity practitioners to enhance penetration test planning efficiency, ensure adequate coverage of critical threat testing, and streamline collaboration with third-party testers. Researchers can leverage the proposed methodology and the MITRE ATT&CK to STRIDE mapping to develop detailed ICS testing procedures, thereby contributing to the advancement of structured ICS security testing practices.

1. Introduction

Industrial Control Systems (ICS) are complex systems of systems that integrate Information Technology (IT) and Operational Technology (OT). Given their vital role in critical infrastructure, securing ICS against cyber attacks remains a persistent challenge (Stouffer et al., 2023). Since the discovery of Stuxnet in 2010, the adoption of secure-by-design principles has become essential in ICS development, with threat modeling playing a key role in the design phase. Despite growing research interest in ICS threat modeling, there remains a lack of readily deployable and empirically validated approaches tailored to ICS environments (Khalil et al., 2024). Moreover, validating threat modeling outcomes and ensuring that selected security requirements are effectively implemented remain significant challenges.

In parallel, ICS penetration testing presents distinct technical and organizational challenges. The integration of IT and OT components results in a broad and complex attack surface, requiring expertise across multiple domains. Unlike IT systems, ICS operate under strict availability and safety constraints, making thorough penetration test planning essential to avoid service disruption or physical damage (Stouffer et al., 2023). Given the potential real-world consequences of ICS attacks, including risks to human safety, ICS penetration testing must consider not only technical expertise, but also the impact and consequences of each selected test.

Threat modeling-based penetration testing methodologies, where threat modeling supports penetration test planning, can help ensure more comprehensive testing and provide a clear rationale for test selection (Süren et al., 2023). Although several threat modeling-based penetration testing methodologies have been proposed for Cyber-Physical Systems (CPS) (see Section 2), they are primarily designed for Internet of Things (IoT) systems, leaving a significant gap in addressing the unique challenges of ICS. While IoT and ICS share common attack surfaces, they differ significantly in system complexity, safety constraints, and deployment context. Consequently, IoT-focused penetration testing methodologies do not adequately address the planning needs of ICS penetration testing.

Furthermore, existing CPS threat modeling-based penetration testing studies predominantly focus on laboratory environments or

* Corresponding author.
E-mail addresses: Shaymaa.Khalil@taltech.ee (S.M. Khalil), Hayretdin.Bahsi@taltech.ee (H. Bahsi), tarmo.korotko@taltech.ee (T. Korõtko).

https://doi.org/10.1016/j.cose.2026.104906
Received 9 December 2025; Received in revised form 13 March 2026; Accepted 18 March 2026
Available online 23 March 2026
0167-4048/© 2026 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

controlled testbeds, which limits the consideration of real-world project constraints and deployment conditions (Süren et al., 2023; Salzillo et al., 2020), revealing a clear gap in real-world case studies that consider project-specific constraints and practical deployment conditions. This gap highlights the need for methodologies that address the complexities of real ICS projects and connect the various development stages, such as incorporating design-phase threat modeling insights to support penetration test planning, as demonstrated in this study.

Resource and time limitations may necessitate various prioritization decisions in the penetration testing process for resource-constrained projects that involve the development of new system components, such as the microgrid system described by Khalil et al. (2023), where a control system was developed. In such contexts, stakeholders might decide to exclude one or more of the four International Electrotechnical Commission (IEC) 62443 component security testing processes (see Section 2) for the newly developed components. Consequently, penetration testing becomes particularly critical, as it indirectly evaluates the effectiveness of the implemented threat mitigation techniques by attempting to exploit the system's vulnerabilities.

A previous study (Khalil et al., 2023) introduced a CPS threat modeling methodology intended for application during the early stage of the design phase, when detailed information about the system software and vendors is typically unavailable. For this reason, a high-level threat modeling method, STRIDE (Howard and Lipner, 2006), was adopted to analyze potential system threats at a high level of abstraction. In the first case study (Khalil et al., 2023), STRIDE proved effective in establishing a clear, understandable common language for cybersecurity specialists and non-cybersecurity experts (e.g., software and power engineers) involved in the threat modeling exercise. The choice of a non-sophisticated threat modeling method, similar to STRIDE, during the design phase facilitated brainstorming and discussions about threat impacts and consequences. However, following system deployment and during planning of the testing phase, discussions of test scenarios require a more detailed framework, such as MITRE ATT&CK for ICS (Alexander et al., 2020), to support planning potential tests.

In addition to the previously discussed challenges, vulnerability ranking in ICS presents a distinct hurdle for effective penetration testing. Traditional vulnerability ranking methods, originally designed for IT environments, often overlook the potential physical consequences that may result from exploiting a vulnerability in an ICS (Stouffer et al., 2023). This gap is significant, as a vulnerability rated as low or medium severity in conventional IT terms may lead to severe operational or safety consequences when exploited in an ICS. For example, a buffer-handling flaw causing a controller reboot after receiving malformed packets may be scored as medium severity in IT terms, yet in an ICS, it could disrupt safety-critical communication and potentially trigger physical incidents. This mismatch between traditional ranking methods and the unique risk profile of ICS environments necessitates the use of more comprehensive ranking systems that incorporate physical impact and safety considerations.

Although the Common Vulnerability Scoring System (CVSS) v4.0 (FIRST, 2023) was released to address several limitations of earlier scoring approaches, its effectiveness in capturing the distinct risk profile of ICS vulnerabilities, particularly when compared to CVSS v3.1, has not yet been thoroughly evaluated through real-world ICS case studies. Furthermore, the limited availability of practical, threat modeling-based penetration testing methods for ICS, especially those leveraging insights from design-phase threat modeling, highlights the need for further research.

Based on the hypothesis that penetration testing outcomes can serve as a validation mechanism for threat modeling conducted during the system design phase, and to address the identified research gaps, this study explores the following two research questions:

RQ1: How can design-phase threat modeling guide penetration test planning for newly deployed industrial control systems?

RQ2: How effectively do vulnerability scoring methods capture the physical and safety-related consequences of vulnerabilities in industrial control systems?

Although the two research questions address different stages of the penetration testing process, they are complementary. RQ1 focuses on structuring and justifying test selection during the penetration test planning phase, while RQ2 focuses on vulnerability prioritization during the vulnerability ranking phase, which follows exploitation. Together, the two research questions support a more systematic, transparent, and traceable penetration testing workflow for ICS.

To address these research questions, this study extends the case study presented by Khalil et al. (2023) and proposes the **B**ridging **R**equirements and **I**mplementation through **D**esign-**G**uided **E**valuation (BRIDGE) methodology, a nine-stage methodology for threat modeling-driven penetration testing of newly deployed ICS. The methodology builds upon the results of the threat modeling conducted during the design phase to guide penetration test planning and prioritization. The proposed methodology is validated through a real-world microgrid case study, in which both CVSS v3.1 and v4.0 were applied for vulnerability ranking.

This paper makes several key contributions to the field of ICS security:

(1) It introduces BRIDGE, a nine-stage threat modeling-driven penetration testing methodology for newly deployed ICS. The methodology connects the design and testing phases by leveraging the outcomes of design-phase threat modeling to guide test selection and prioritization.

(2) It extends and formalizes a mapping of MITRE ATT&CK for ICS techniques to STRIDE threat categories. This mapping bridges the gap between high-level threat analysis and intermediate-level attack representations required for penetration test planning, enabling the development of comprehensive and well-justified testing scenarios.

(3) It evaluates the applicability of CVSS v4.0 compared to CVSS v3.1 in representing the distinctive risk profile of ICS vulnerabilities, particularly in terms of safety implications and security requirements.

(4) It presents a real-world case study of ICS penetration testing using a newly deployed microgrid system, providing valuable insights into the practical complexities and challenges encountered in real-world projects, which are often absent in controlled laboratory environments. These insights can help practitioners better understand the realistic constraints and decision-making processes involved in ICS penetration testing.

(5) Finally, it demonstrates how penetration testing can serve as a validation mechanism for the design-phase threat modeling outcomes, confirming that the identified requirements were correctly applied and that the system is resilient against the prioritized threats.

A preliminary version of this work was presented as a conference paper (Khalil et al., 2026). The present manuscript substantially extends that work by formalizing and naming the BRIDGE methodology, integrating the existing methodology steps into a coherent threat modeling-driven penetration testing framework, and providing an expanded case study, analysis, and discussion.

The rest of this paper is organized as follows: Section 2 discusses the background and related work; Section 3 explains the research design and introduces the BRIDGE methodology for threat modeling-driven penetration testing of ICS. Section 4 presents the results of the microgrid case study, structured according to the stages of the BRIDGE methodology. Section 5 discusses the implications of the results, while Section 6 examines the threats to the validity of the results. Finally, Section 7 concludes the study.

2. Background and related work

To ensure a secure ICS product development lifecycle, the IEC 62443-4-1 standard (Commission, 2018) defines eight key practices for secure product development. These include specifying security requirements (through threat modeling), applying secure-by-design principles (which

Table 1
IEC 62443 cyber security testing stages, as described by Commission (2018, 2022).

Cyber Security Testing Stage	Responsible Entity	Objective	Security Verification and Validation Testing Processes
Component testing	Vendor	Verify that each component meets its operational and security requirements	Security requirements testing Threat mitigation testing Vulnerability testing Penetration testing
Integration testing	Integrator	Verify that all components work together as designed	Non-production integration tests
System testing	Might be run by third party under the system owner responsibility	To demonstrate that the system management, operational, and technical countermeasures are implemented correctly, effective, and meet the requirements.	Penetration testing Policies and procedures testing Manual security checks

involve describing how the design mitigates the identified threats), and conducting security verification and validation testing. The latter is recommended at the three stages of ICS cyber security[1] testing: component, integration, and system testing, as shown in Table 1, and should be verified by the system owner. Verification testing aims to confirm that "you built it right", for example, by verifying that the implemented system meets its defined security requirements. On the other hand, validation testing aims to ensure that "you built the right thing", for example, by assessing whether the implemented system can withstand the prioritized threats identified during the threat modeling process (Commission, 2022).

As part of security verification and validation practices for a developed ICS component, the IEC 62443-4-1 (Commission, 2018) identifies four distinct testing processes: security requirements testing, threat mitigation testing, vulnerability testing, and penetration testing. However, it does not provide detailed guidance on how to effectively integrate and harmonize these testing efforts.

The security requirements testing process, which can be categorized as verification testing, aims to ensure that the security functions of the newly designed ICS component meet its security requirements. This testing involves various approaches, including functional testing, performance testing, and unexpected input tests to assess the product's behavior under different scenarios. However, the threat mitigation testing process verifies that the threat mitigation techniques function as designed and validates their effectiveness in protecting the system component against the threats identified by the threat modeling. For example, variant authentication tests might be used to ensure that the implemented authentication mechanism effectively mitigates elicited spoofing threats (Commission, 2018).

The vulnerability assessment process focuses on identifying security weaknesses within an ICS component. It implies different types of testing, including abuse-case testing and known vulnerability scanning. Vulnerability assessments typically do not involve exploiting the identified vulnerabilities. However, known vulnerability scanning serves as a critical foundation for penetration testing activities. Penetration testing, in contrast, adopts a more proactive approach. It simulates real-world cyberattacks by attempting to exploit identified vulnerabilities to assess the overall security posture of a system. This enables organizations to identify and mitigate security gaps before they are exploited by malicious actors.

The IEC 62443 standard does not provide a clear description of the processes involved in the verification and validation of integration and system testing. Instead, it describes examples of processes that might be used in these two stages, as shown in Table 1. It is important to note that penetration testing was mentioned in both component and system testing. Penetration testing may serve as both verification and validation, as it verifies that the system is implemented as designed and validates that it is resilient to the threats mitigated during the threat modeling in the design phase.

ICS penetration testing overlaps with IT penetration testing in tasks similar to operating system penetration testing (e.g., Windows and Linux), as well as web application testing (Fujimoto et al., 2023). However, ICS penetration testing requires additional knowledge, as such systems use specialized protocols and standards that are not used in IT. The scope of IT penetration testing is usually limited to cyberspace, whereas attacks on ICS may result in physical impacts that should be considered during penetration testing. Also, IT system penetration testing primarily focuses on confidentiality breaches, while ICS prioritize process availability and data integrity, which should be reflected in the testing scenarios (Fujimoto et al., 2023).

Electricity is a life-sustaining service, and its disruption can significantly disrupt daily life. For this reason, infrastructure related to electricity generation and transmission was among the first critical infrastructures identified by the European Union (Directive, 2008). To the best of the authors' knowledge, no official guideline currently exists within the European Union for the penetration testing of electric utilities. However, such a guideline has been proposed by a United States-based organization, the National Electric Sector Cyber Security Organization Resource (NESCOR) (Searle et al., 2016).

NESCOR guidelines break the penetration testing tasks of an electrical utility into five stages: *penetration test scoping, architecture review, target system setup, end-to-end penetration test analysis*, and *result interpretation and reporting*. The guide categorizes penetration testing tasks into four categories based on the required skills and the time required to perform each task. The *server Operating System (OS)* penetration tasks require basic testing skills, should be performed frequently, and are mostly done by internal teams. The *server application* penetration testing tasks require moderate testing skills and are commonly performed. The *network communication* penetration tasks require greater expertise, making them occasionally performed. The *embedded device* penetration tasks are advanced tasks that require highly specialized skills, and they are time-consuming (Searle et al., 2016).

While NESCOR does not explicitly refer to "threat modeling", its recommendation to review the target system architecture through stakeholder interviews helps understand the system's functionality, intended security posture, and the potential security risks associated with vulnerabilities. This approach aligns with the core objectives of threat modeling. In contrast, other penetration testing methodologies, such as the Penetration Testing Execution Standard (PTES) (Penetration Testing Execution Standard, 2026), explicitly incorporate threat modeling as a distinct phase. PTES defines seven stages for penetration testing: *pre-engagement interactions, intelligence gathering, threat modeling, vulnerability analysis, exploitation, post-exploitation*, and *reporting*.

[1] In this paper, the term cybersecurity is used in accordance with contemporary academic usage. However, when referring to official terminology from standards or organizations (e.g., IEC 62443 or the National Electric Sector Cyber Security Organization Resource), the original spelling cyber security is preserved for consistency.

Threat modeling-based penetration testing has also been proposed in several academic studies; however, most of these methodologies primarily focus on IoT systems. For example, Salzillo et al. (2020) proposes a four-stage methodology for IoT systems, consisting of *system modeling, threat modeling, planning, and penetration testing*. The study applies the proposed methodology to a home automation system, with a particular focus on threat modeling for the Message Queuing Telemetry Transport (MQTT) protocol,[2] which constitutes the paper's main contribution. It employs the Multi-Cloud Application Composition Model (MACM), a graph-based system modeling, to support threat elicitation using a threat catalog and the Common Attack Pattern Enumeration and Classification (CAPEC) (MITRE Corporation, 2023). The study provides detailed descriptions of each attack pattern of potential system threats, which are then used to formulate an effective attack plan.

Practical and agile threat research for IoT (PatrIoT) is a four-stage IoT penetration testing methodology (Süren et al., 2023). The *planning, threat modeling, exploitation*, and *reporting* stages are further divided into well-documented substages. PatrIoT decomposes an IoT system into seven attack surfaces: *hardware, firmware, network, web, cloud, mobile*, and *radio*. The methodology compiles the Open Worldwide Application Security Project (OWASP) IoT Top 10 (OWASP, 2023) into a list of 100 security weaknesses based on the Common Weakness Enumeration (CWE) (MITRE Corporation, 2026). PatrIoT is designed primarily for the penetration testing of IoT products, including drones, smart fridges, and surveillance cameras.

Performing threat modeling within the penetration testing process is often a brief exercise focused on helping testers understand the system. It typically does not involve developing an extensive list of threats and often lacks information about system security requirements or the system's security posture. Design-phase threat modeling systematically addresses this by performing comprehensive threat analysis, threat prioritization, and security requirements selection for the most critical threats. However, existing penetration-testing methodologies typically treat threat modeling as an activity conducted only during the testing stage. This might be related to the typical timing of penetration testing, which is often performed long after system deployment, when secure-by-design practices may not have been applied or not documented in a way that testers can use. Consequently, the connection between early threat modeling results and penetration testing activities remains weak, leaving a methodological gap in how threats inform test selection.

Threat prioritization is a crucial stage in threat modeling conducted during the design phase. It helps justify the selection of mitigation techniques by identifying the most critical threats (Khalil et al., 2023). Similarly, categorizing vulnerabilities identified during penetration testing is essential. Vulnerability ranking facilitates prioritizing mitigation efforts and can be carried out using vulnerability assessment methods, which are typically categorized as quantitative or qualitative. Quantitative vulnerability assessment methods, such as CVSS, assign numerical scores to vulnerabilities. Whereas qualitative methods classify vulnerabilities into severity classes (e.g., high, medium, low) (Sharma et al., 2023).

CVSS is a vendor-agnostic "*open framework for communicating the characteristics of software vulnerabilities*" (FIRST, 2023). It has multiple versions: CVSS v2 and v3 are widely adopted, while CVSS v4.0 is the most recent release. CVSS v4.0 calculates its score based on four metric groups: *base*, *threat*, *environmental*, and *supplemental*. The *base metric* group comprises exploitability and impact metrics that remain constant over time and across user environments. The *threat metric* group may change over time, as it depends on the exploit maturity. The *environmental metric* group is related to vulnerability characteristics that depend on the user environment, similar to applied security controls. The *supplemental metric* group is defined by the user and considers additional factors similar to the impact of the vulnerability on system safety and its resilience to recovery from the vulnerability exploit, which enhances the relevance of CVSS v4.0 for ICS environments.

Published CVSS scores are typically limited to base scores, as they are independent of customer environments. Organizations are expected to adjust these scores using the remaining metrics to produce context-specific CVSS scores. Vulnerability assessment in ICS and OT systems must consider both the potential safety impact and the system's resilience within its specific environment, factors that were not adequately addressed in earlier CVSS versions.

Several studies have applied structured threat analysis methods and adversary behavior frameworks, such as STRIDE (Howard and Lipner, 2006) and MITRE ATT&CK (Alexander et al., 2020), in ICS security research. For example, Choi et al. (2020) build an ICS testbed and use the MITRE ATT&CK framework to identify monitoring fields for generating attack datasets. They further introduce a data enrichment tool to expand the collected datasets by generating synthetic events that simulate large-scale and long-duration attack scenarios. Similarly, Tharot et al. (2023) employ ATT&CK-based game scenarios for industrial cybersecurity training, and Da Silva et al. (2023) propose an automated STRIDE-based template for the Microsoft Threat Modeling Tool in ICS contexts. While these works demonstrate the applicability of STRIDE or ATT&CK to ICS environments, they primarily focus on testbed development, training, or tooling. In contrast, this work addresses the systematic integration of design-phase threat modeling results into penetration test planning for newly deployed ICS.

Planning industrial control systems penetration testing is a challenging task due to their complexity and the wide range of technologies they often involve. Defining the testing scope and planning tests with threat modeling guidance would help identify testing priorities and support a more structured testing process. Although a limited number of studies have proposed threat modeling-based penetration testing approaches for CPS, these works typically apply threat modeling during the penetration testing stage to elicit test cases (Süren et al., 2023; Salzillo et al., 2020). However, they overlook the value of carrying forward the results of threat modeling performed during the design phase into later testing activities. Incorporating early-stage outputs, such as system models, elicited threats, and prioritization results into subsequent security activities could improve both the efficiency and effectiveness of the testing process. Moreover, there is a notable scarcity of real-world case studies that document threat modeling-based penetration testing in ICS contexts.

3. Methodology

To address the identified research gaps, this study proposes a threat modeling-driven penetration testing methodology referred to as **BRIDGE** . The methodology leverages the outcomes of the threat modeling conducted during the design phase to guide penetration test planning before the system transitions into operational use. The term threat modeling-driven is deliberately used, rather than threat modeling-based, to emphasize that the comprehensive threat modeling conducted during the system design phase plays a central role in shaping the penetration testing strategy. This approach serves as a bridge between the secure ICS design phase and its subsequent verification and validation activities.

Fig. 1 illustrates the bidirectional relationship in the threat modeling-driven penetration testing approach. Existing threat modeling-based penetration testing methods are typically unidirectional, where threat models are created during the penetration testing phase to support test planning. These methods do not incorporate or link the threat modeling efforts performed during the design stage into later testing activities. In contrast, the threat modeling-driven approach incorporates a feedback loop: penetration testing results are used to verify that the specified security requirements were correctly implemented and to validate that the threats identified during threat modeling have been effectively

[2] A lightweight publish-subscribe messaging protocol widely used in IoT systems.

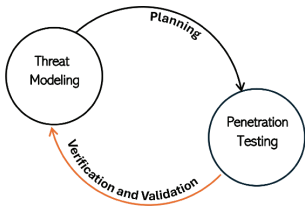


Fig. 1. The relationship between threat modeling and penetration testing in the proposed threat modeling-driven penetration testing methodology.

mitigated through the chosen requirements. This bidirectionality ensures that threat modeling is not only a planning tool but also a living artifact that is validated and refined based on testing outcomes, thereby strengthening the alignment between design-phase threat modeling and the actual security posture of the deployed system.

Fig. 2 illustrates the BRIDGE methodology for penetration testing, which consists of nine stages: (i) ICS Threat Modeling, (ii) Data Flow Diagram Update, (iii) Penetration Test Scoping, (iv) Penetration Test Planning, (v) Exploitation, (vi) Vulnerability Ranking, (vii) Reporting, (viii) Clean-up, and (ix) Report Analysis, Remediation and Retesting.

Unlike threat modeling-based penetration testing methodologies (see Section 2), the first stage of the BRIDGE methodology, ***ICS Threat Modeling***, is initiated during the design phase rather than the testing phase. Conducting this stage early in the lifecycle allows the identification of potential threats before system deployment and enables the definition of critical security requirements. This stage corresponds to the comprehensive threat modeling activities performed during the system design and relies on the structured methodology for CPS threat modeling introduced in Khalil et al. (2023). The methodology provides a systematic process for identifying information and system assets, mapping the system into a Data Flow Diagram (DFD), defining the security context, determining trust boundaries, eliciting threats, analyzing their consequences and potential losses, prioritizing threats based on impact analysis, and finally deriving the system's critical security requirements. The resulting threat model forms the cornerstone of the BRIDGE methodology, supporting subsequent stages by linking high-level design insights to practical penetration testing objectives and scope definition.

The ***Data Flow Diagram update*** stage is initiated during the system implementation and involves discussions related to the system testing procedures. These discussions address whether component, integration, and system tests will be conducted separately, as well as the selection of the security verification and validation processes to be applied to the newly developed system component(s) (i.e., security requirement testing, security mitigation testing, vulnerability testing, and penetration testing). A decision regarding whether the tests will be executed internally or by a third party should be made as early as possible to allow sufficient time for external arrangements, if required. The DFD is updated at the conclusion of the implementation phase to ensure that no further changes to the system are introduced before testing. Following the system implementation, the cybersecurity team revises the previously identified threat list, including prioritization. It is also recommended at this stage to document both the implemented and excluded security requirements, as this facilitates test selection and supports follow-up during penetration testing result analysis.

The ***Penetration Test Scoping*** stage defines the boundaries and objectives of the penetration test. During this stage, stakeholders agree on the testing type (white-box, gray-box, or black-box) based on testing goals. This study advocates the use of white-box testing, particularly when the internal cybersecurity team that participated in the threat modeling phase also conducts the testing. In such cases, the process becomes more efficient and less time-consuming. Conversely, gray-box testing is often more suitable when testing is performed by external teams. Providing external testers with limited yet essential information, such as high-level system architecture and critical information flows, enables them to concentrate on the most relevant attack vectors. While white-box testing offers deeper insight and traceability, it may present challenges for external engagements due to time constraints, confidentiality or intellectual property concerns, and higher costs associated with extended engagements.

During the scoping stage, stakeholders may agree on the complexity level of the penetration testing tasks to be included in the tests (Searle et al., 2016). This decision can be guided by the types of system components and attack surfaces, discussed in Section 2. In this study, the categorization proposed by NESCOR (Searle et al., 2016) is followed, which identifies four penetration testing task categories for electrical utilities.

The *Server OS penetration tasks* involve vulnerability assessment of Windows and Linux-based systems and aim to identify exploitable system vulnerabilities that could lead to unauthorized access. As described by Searle et al. (2016), these tasks comprise three stages: information gathering, vulnerability analysis, and exploitation. Penetration testers with limited expertise in ICS are usually familiar with these tasks, and a wide range of tools support these tests (e.g., Metasploit Rapid7, 2026).

The *Server Application penetration tasks* focus on testing applications to discover their security weaknesses. The testing activities include application mapping, application discovery, and application exploitation (Searle et al., 2016). These tests are generally more complex and time-consuming than the server operating system tasks.

The *Network Communication penetration tasks* aim to exploit vulnerabilities in network protocols and communication mediums. They comprise two subcategory tasks: radio frequency packet analysis and network protocol analysis. These tasks are complex, as they require in-depth knowledge of network protocols and may involve specialized tools, such as those used for wireless attacks. This is particularly relevant, as ICS environments, including electric utilities, rely on diverse wireless technologies, such as Wi-Fi, cellular networks, and the Global Positioning System (GPS). Protocols such as Zigbee (ZigBee Alliance, 2015), Modbus (Modbus Organization, 2012), and DNP3 (DNP Users Group, 2005) are also widely used in ICS networks, and attacks against them require a solid understanding of protocol datagrams and functionality.

Embedded Device penetration tasks include electronic component analysis, field technician interface analysis, and firmware binary analysis, and are the most demanding and time-consuming penetration testing tasks. They involve testing embedded and microprocessor-based devices, including firmware, input interfaces such as serial and parallel ports, as well as non-volatile data storage such as flash memory and EEPROM, which typically store firmware and configuration data (Searle et al., 2016).

Deciding which task categories to include in the testing scope sets expectations regarding the testing timeframe, required expertise, budget, and expected outcomes. Excluding one of the mentioned four task categories implies accepting the possibility of undiscovered vulnerabilities within that category and requires explicit approval from the relevant stakeholders.

Another key decision during the scoping stage involves selecting which threats to include in the testing plan. Threats identified during the threat modeling exercise and updated during the *DFD update* stage often form a long list of threats that vary in severity and consequences. Selecting a prioritized subset of threats for testing is therefore essential for a focused and effective penetration test. This prioritization should be based on explicit criteria, such as criticality or expected impact type, so that testers can concentrate on the most consequential threats and optimize the testing effort and its outcomes.

The scoping stage should also include discussions of the targeted domains and IP ranges, the specific systems and environments to be tested, and the level of access to be granted. It should define the types of attacks to be included or excluded from testing, such as DoS and social engineering. Since the tests are conducted before the system is

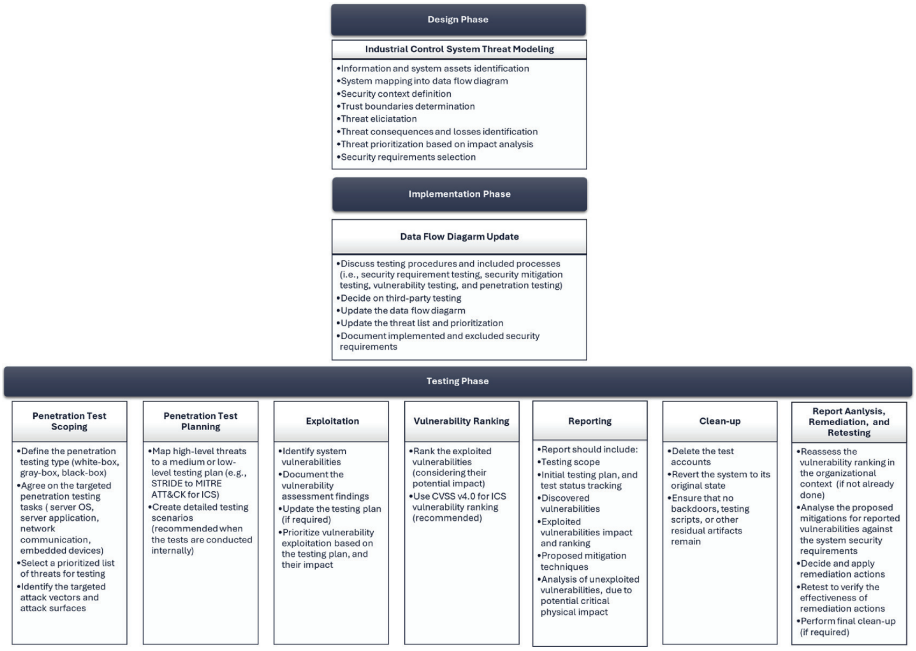


Fig. 2. BRIDGE: A threat modeling-driven penetration testing methodology for newly deployed industrial control systems.

transitioned to operations, social engineering attacks might not be considered, as no operators are yet involved. For DoS testing, the decision to include or exclude such tests should depend on their potential consequences, as some tests may cause service interruptions or component failures even in a non-operational environment. In addition, the scope should clearly identify the attack surfaces relevant to the ICS environment, such as remote access interfaces, internet-facing applications, and potential lateral movement paths within the network. These discussions help establish realistic testing boundaries and minimize the likelihood of unintended disruptions.

The ***Penetration Test Planning*** stage is carried out based on the previously defined testing scope. During this stage, testers may rely on supporting frameworks and reference guides to structure the testing plan. Guides, such as the NESCOR testing guide (Searle et al., 2016) and STRIDE threat trees (Howard and Lipner, 2006), provide high-level guidance for the test planning without detailed testing procedures. More detailed procedures are available in domain-specific resources such as the OWASP testing guides (OWASP, 2026), which cover web security, mobile application, and firmware security testing methodology.

In practice, penetration test planning approaches vary across organizations and projects and are strongly influenced by operational, safety, and availability constraints, particularly in environments involving operational technology (Stouffer et al., 2023). Existing guidance emphasizes that all penetration testing engagements should begin with careful planning and scoping to ensure that testing activities do not adversely impact operational systems (Searle et al., 2016). A key contribution of this study is to demonstrate how an ICS penetration testing plan can be guided by STRIDE threats mapped to MITRE ATT&CK for ICS techniques (see Section 4.3.2), supporting structured and risk-informed test selection. Depending on the agreed-upon timeframe and available resources, testers may prepare a detailed plan that defines testing scenarios and procedures. Alternatively, when testing is performed by third-party teams, as in the case study presented in Section 4, the planning stage may result in a high-level plan that outlines the techniques to be tested while leaving the detailed technical procedures to the tester.

The ***Exploitation*** stage is guided by the testing plan. Testers identify system vulnerabilities and execute the planned tests while prioritizing the exploitation based on the plan and potential impact. They document their findings throughout the process, updating the testing plan as needed to address newly discovered attack vectors. During the ***Vulnerability Ranking*** stage, exploited vulnerabilities are analyzed and prioritized based on their potential impact. This may involve using CVSS or a custom ranking system tailored to the ICS environment. Based on the case study results, presented in Section 4, this study recommends using CVSS v4.0 for ICS vulnerability ranking due to its consideration of the impact of the vulnerability on the system safety. The vulnerability ranking stage ensures that the mitigation efforts focus on the most critical vulnerabilities.

Testers should document all findings in a comprehensive report during the ***Reporting*** stage. The report may follow the OWASP guide (Saad and Mitchell, 2020) by mentioning the test scope, limitations, timeline, and an executive summary of the test objectives, key findings, and recommendations. The findings section should also follow the best practices (Saad and Mitchell, 2020), and include: Vulnerability type and rating, impact, root cause (e.g., code bug, configuration issue, security flaw), used testing technique(s), and remediation. The successful tests should be documented in detail, allowing the reproduction of the test results. In the context of threat modeling-driven penetration testing, this study recommends that the report include a briefing about the initial test plan (which may be added as an appendix) and highlight which tests were successful, failed, or excluded due to time limitations or unfulfilled preconditions.

For systems that require an IEC 62443 security level four,[3] It is further recommended to report all discovered vulnerabilities,

[3] "Prevent the unauthorized disclosure of information to an entity actively searching for it using sophisticated means with extended resources, Industrial

including those that were not exploited during the testing period. The report should also include an analysis of critical unexploited vulnerabilities, as in ICS environments, some vulnerabilities may pose a significant risk of physical damage to equipment or have human safety implications. In such a case, testers should prioritize safety by avoiding full exploitation of the vulnerability. Instead, a proof-of-concept demonstration should be conducted to validate the vulnerability and provide sufficient technical details to help stakeholders understand its potential impact, consequences, and possible mitigation measures.

When penetration testing is outsourced to a third party, it may be difficult to obtain the level of detail recommended by this methodology in the final report. However, clearly defining the expected content and level of detail during the initial negotiations with the testers can help ensure that the final report includes the required level of information. The three stages (***Exploitation***, ***Vulnerability ranking***, and ***Reporting***) can be executed in parallel or sequentially, depending on the project's complexity and available resources.

Following the reporting stage, it is essential to initiate a ***Clean-up*** stage to ensure the test environment is fully restored. All accounts created for testing purposes should be deleted, system configurations should be reverted to their original state, and any files or artifacts left on compromised assets should be removed. Placing the clean-up stage after the reporting stage allows stakeholders time to verify and review the report findings before the system is reset. This sequencing is consistent with best practices recommended in the professional penetration testing literature,[4] alongside the subsequent remediation and retesting stage.

The BRIDGE methodology concludes the penetration testing process with the ***Report Analysis, Remediation, and Retesting*** stage, where the reported vulnerabilities are re-assessed within the context of the organization, particularly if contextualization was not fully addressed during the reporting stage. If penetration testing is prioritized over security requirements testing and threat mitigation testing within the verification and validation processes, then the cybersecurity team should analyze the proposed mitigations for reported vulnerabilities, identified through penetration testing, against the system security requirements. Stakeholders may also review the testing plan status to check whether tests were successful, failed, initially excluded, or excluded during the exploitation stage due to time or technical limitations. This review helps the stakeholders identify whether any threats from the prioritized list were not tested during the exploitation stage.

Remediation actions for the reported vulnerabilities should then be identified and implemented, followed by verification and validation testing to ensure the applied measures are effective. Retesting is especially critical for high and critical severity vulnerabilities, as they may lead to a significant impact if left unresolved before the system becomes operational. If necessary, a final clean-up round should be conducted after retesting to remove any temporary remediation or diagnostic elements introduced during the testing process.

The BRIDGE threat modeling-driven penetration testing methodology provides a structured approach for ICS penetration testing, bridging the design and testing phases. By addressing the unique ICS challenges through threat modeling, vulnerability prioritization, and comprehensive reporting, the methodology enhances the effectiveness and efficiency of security assessments, ultimately contributing to a stronger security posture for ICS.

4. Microgrid case study

This study demonstrates the application of the BRIDGE threat modeling-driven penetration testing methodology through a real-world case study of a newly developed microgrid system. Since the threat modeling for this microgrid, conducted during the design phase, was discussed in detail in Khalil et al. (2023), this paper focuses on the results of the proposed penetration testing methodology, starting from the second stage: Data Flow Diagram Update.

Sections 4.1–4.3 collectively address RQ1 by presenting the results of applying the BRIDGE methodology to the microgrid case study. Section 4.5 addresses RQ2 by comparing the severity ranking of the reported vulnerabilities using CVSS v3.1 and CVSS v4.0. Finally, Section 4.8 evaluates the research hypothesis defined in Section 1 by validating it against the results of the case study.

The penetration testing process involved a collaborative effort among five stakeholders: the cybersecurity team (including the first and second authors); the power systems expert (the third author), representing the power engineering team and the system owner; the software architect; and the third-party penetration tester. The penetration testing process was conducted over eight months, beginning in February 2023 and concluding in September 2023. During this period, stakeholders participated in 13 technical meetings to discuss the details of each stage. These meetings did not include managerial discussions related to the selection of the testing entity or contractual agreements.

4.1. Data flow diagram update

As a reminder of the microgrid system introduced in Khalil et al. (2023), the hardware structure of the analyzed system includes a microgrid control unit, mapped as the Server in the DFD. This unit consists of computational equipment and a software platform and is responsible for data and log management, as well as other services such as monitoring. The Server is based on an in-house software platform designed and implemented by the project's research teams.

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 utilization of locally produced energy and to provide capacity management, as well as ancillary services such as power quality improvement, voltage support, and system restoration. The Moulded Case Circuit Breaker (MCCB) is a power system component required to connect the microgrid to the Distribution System Operator (DSO) grid through the Point of Common Coupling (PCC). The system further includes PhotoVoltaic (PV) generation and may facilitate one or more prosumers, which are proactive entities that either produce or consume energy.

In the DFD presented in Khalil et al. (2023), energy consumption was measured using energy meters, while operational measurements of power system properties were performed using multimeters. Meter data concentrators and PV data concentrators, implemented as in-house software components, were used to collect and aggregate data from different parts of the system and communicate it to the Server.

During the DFD updating stage, the cybersecurity team collaborated with the software architect and the power systems expert to discuss design changes implemented since the initial DFD creation. These discussions revealed several system design modifications, which were reflected in the updated version of the DFD, shown in Fig. 3. Compared to the DFD created during the design phase and discussed in Khalil et al. (2023), all energy meters were replaced with multimeters. Unlike an electric energy meter, a multimeter is capable of measuring multiple parameters and enables both energy metering and the measurement of electrical properties such as voltage and current. Consequently, the "meter data concentrator", which was intended to host an in-house software application for collecting data from energy meters, was removed.

The updated DFD also reflects a refinement of the trust-zone definitions introduced during the design-phase threat modeling. As a result of these refinements, Trust Zone 3 was eliminated, and the prosumer multimeter was relocated from Trust Zone 1 to Trust Zone 2. In addition, a new Trust Zone 4, representing the OT network, was added as a result

Automation and Control System (IACS) specific skills and high motivation" (Commission, 2019).

[4] https://www.coresecurity.com/blog/six-stages-penetration-testing

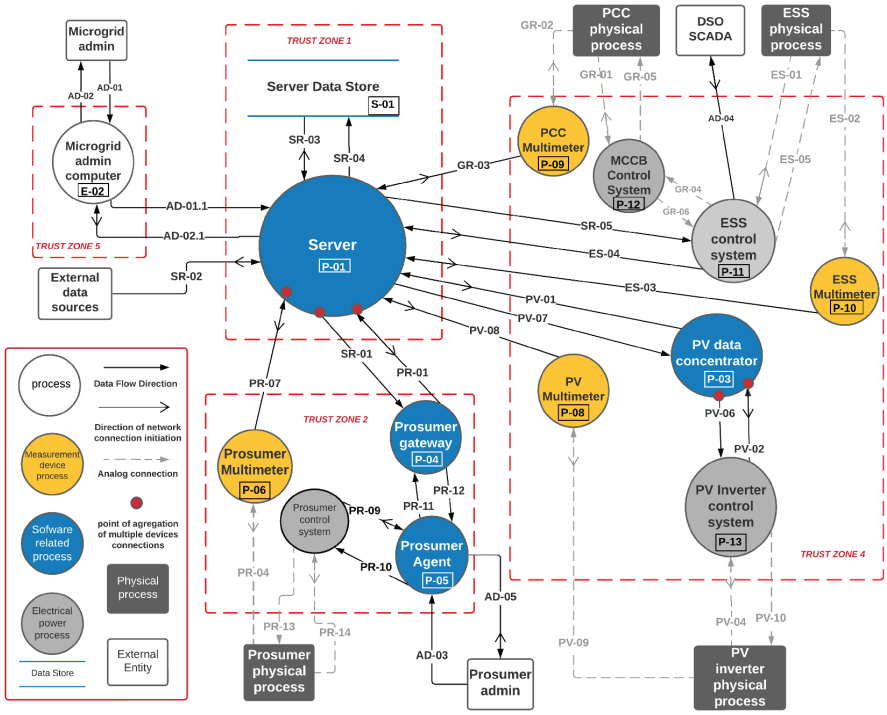


Fig. 3. The updated microgrid data flow diagram.

of the implemented segmentation. Furthermore, in the initial DFD, the microgrid administrator computer was modeled as an external process. However, this representation was corrected in the updated DFD, as the microgrid administrator workstation is managed by the same organization operating the microgrid and does not constitute an external entity in the strict sense. Instead, the administrator computer was placed in a separate trust zone to reflect a different security posture and access pattern, while still being treated as an internal process under organizational control.

The DFD updates significantly impacted the list of 110 threats identified during the design phase. The changes resulted in the suppression of 19 threats deemed irrelevant and the introduction of three new threats. Subsequently, the elicited threat list was refined, and the prioritization was re-evaluated based on these updates.

Considering resource and time constraints, the stakeholders prioritized penetration testing within the IEC 62443 security verification and validation testing processes. As a result, security requirements and threat mitigation tests were excluded from the project. Negotiations to determine whether the penetration testing would be conducted internally by the cybersecurity team or outsourced to a third party were concluded during the system implementation phase.

The stakeholders decided to involve a third-party penetration tester, ensuring a fresh perspective on the system's security posture. Engaging a third party with specialized penetration testing experience and skills aimed to support the effective management of the limited testing time. This approach ensured better resource management, given the limited number of internal cybersecurity experts involved in this project.

Negotiations with the third-party penetration tester proceeded in parallel with the DFD updates. The final agreement was reached with a professional freelance penetration tester with extensive experience in IT system penetration testing and a working background in OT systems. Once the agreements and DFD updates were finalized, the tester was provided with the updated DFD, data flows, and the prioritized threat list. Sharing the threat modeling outputs at this stage was intended to support the tester's understanding of the system's threat landscape and potential impacts. Follow-up meetings were then scheduled to review the provided documents in detail and address any questions raised by the tester.

To address the penetration tester's specific inquiries regarding the network protocols used within the microgrid system, the data flows mapping table, originally described in Khalil et al. (2023), was updated to include the protocol associated with each data flow, as illustrated in Table 2. Unlike traditional software data flow diagrams, which typically represent data exchange between software processes, an ICS DFD mostly illustrates information flows between physical devices and subsystems. Therefore, annotating the network protocols used for each data flow provided valuable context for the penetration tester, potentially facilitating the identification of vulnerabilities specific to those protocols. The DFD update stage was intended to ensure that all relevant documentation was updated and that the tester fully grasped the data flow diagram and critical system threats.

4.2. Penetration test scoping

Following the documentation updates and the engagement of the penetration tester, stakeholder meetings were held to define the penetration testing scope. The penetration testing type was agreed upon as a gray-box test, where the DFD, high-level network diagrams, and threat lists were provided to the tester, along with addressing the tester's inquiries. However, the source codes were not disclosed.

The stakeholders then discussed the complexity of the penetration testing tasks and agreed to prioritize testing of OS, server applications, and network communication. The scoping decisions adopted for the penetration testing activities were driven by the consequence and loss assessment performed during the design-phase threat modeling, as presented in Khalil et al. (2023). In particular, priority was given to the

Table 2
Example of updated entries in the data flow mapping table, with updates highlighted in bold.

Flow ID	From	ID	To	ID	Information Asset Type	**Network Protocols**
PV-01	PV data concentrator	P-03	Server	P-01	Operational Data*	**Modbus TCP**
PV-02	PV inverter control system	P-13	PV data concentrator	P-03	Operational Data*	**Modbus RTU over RS485**

*Operational Data: data representing the operational status of assets.

in-house developed software components, especially the Server, which functions as the microgrid control unit and hosts internet-facing services. The impact assessment conducted during the design phase (Khalil et al., 2023) showed that spoofing attacks targeting the Server could lead to severe system-level consequences, including exceeding operational limits and misleading safety operations. These consequences may escalate into safety-critical losses, including loss of life or injury to people.

Embedded device testing was deprioritized during this testing phase due to time and resource constraints and because the embedded components were supplied by established ICS vendors that apply baseline security controls. This scoping decision does not reduce the criticality of the findings, as compromise of the in-house software could result in severe safety and availability impacts on the microgrid.

The scoping stage discussions further included a review of the list of critical threats and DFD elements to be included within the testing scope, together with their associated consequences. Potential attack scenarios for the selected DFD elements were brainstormed, considering considering how such attacks would manifest in the system and how the penetration tester could verify their success. The importance of critical threats guiding the testing plan was emphasized while allowing the tester flexibility to explore beyond the predefined threat list.

The meetings then focused on potential attack surfaces (e.g., internet-facing web applications), attack vectors (e.g., unpatched vulnerabilities, weak credentials), threat actors (e.g., malicious actors), and possible attack scenarios. The system architecture places the most safety-critical OT components within a segmented OT network (Trust Zone 4), where they are protected from direct internet access. Accordingly, the penetration testing effort prioritized externally reachable attack surfaces that, based on the impact assessment, could enable an internet-based adversary to influence critical system functions. The tester was first expected to attempt access to the system through publicly accessible interfaces and applications. When this was not feasible within the agreed testing timeframe, controlled access to Trust Zone 1 was provided to allow the assessment to proceed. The OT network itself remained within scope, and the tester was explicitly challenged to attempt to reach it through network-based attacks without being granted direct access.

Discussions also addressed the testing location, ultimately deciding on remote testing while excluding attacks requiring physical presence on-site. Social engineering attacks, such as phishing emails, were also excluded as they rely on user interaction, which was outside the scope of this initial testing phase since the system was not yet operational.

Stakeholders then discussed how the tester would gain initial access to each zone, considering the specific attack vectors relevant to each zone and the security measures in place to limit access. A list of required tester access (e.g., Virtual Private Networks (VPN), jump host) and permission levels (e.g., regular user, admin) was created. To allow the tester to observe the impact of attacks on system alerts, regular user and administrator access were provided to specific tools.

The discussions also addressed the depth of testing, considering whether to focus solely on initial device access or include attempts of lateral movement within zones. Ultimately, it was agreed to provide the tester with limited-access accounts that allowed initial access to selected devices. This approach enables the tester to assess the possibility of lateral movement within the system. Additional discussions included testing time frames, critical vulnerability reporting procedures, and the protocol for handling DoS attacks.

In summary, the penetration test scoping stage defined the testing type, boundaries (i.e., out-of-scope activities), critical attacks to be executed and their consequences, rules of engagement, emergency contacts, and expectations about the penetration testing timeframe.

4.3. Penetration test planning

The cybersecurity team was keen on structuring the planning of the penetration testing based on an attack database that complements the penetration tester's knowledge. However, the selection of a suitable attack database was a major challenge. During the design phase, the microgrid attack taxonomy proposed in Bahsi et al. (2022) was updated. This high-level taxonomy proved valuable for threat elicitation, as described in Khalil et al. (2023). However, it lacked specific details on attack execution, which is crucial for planning penetration testing. For this reason, the use of STRIDE threat trees (Howard and Lipner, 2006) was investigated as an attack database for this study. The threat trees were discussed in detail during two meetings with the penetration tester. While these trees provided a structured approach, their primary focus on software threats created significant gaps in coverage of ICS-specific attacks. This limitation would have resulted in an incomplete penetration testing plan.

STRIDE is considered a high-level threat analysis model (Strom et al., 2018), suitable for threat elicitation during the design phase, where detailed system specifications (e.g., protocols, vendors, OS) are unavailable. However, such information becomes available after system implementation and during penetration test planning. Recognizing the limitations of both the high-level taxonomy, used in Khalil et al. (2023), and STRIDE threat trees, alternative approaches were explored to achieve a more detailed level of modeling that includes richer attack execution details.

As the tests were performed by a third party, the cybersecurity team aimed to guide the testing plan without imposing specific tools and scenarios, as this is typically the responsibility of the tester. For this reason, it was decided to use MITRE ATT&CK for ICS, a mid-level modeling framework that provides details about attack techniques (Strom et al., 2018). However, MITRE ATT&CK for ICS is not designed specifically for penetration testing and lacks detailed information on how to execute the tests. While it identifies attack techniques, the penetration tester would still need to identify the specific tools and exploit each technique.

Additionally, since the threat elicitation was done using STRIDE, mapping these threats to the tactics and techniques of MITRE ATT&CK for ICS became crucial. At this stage, it was decided to have focused meetings between the cybersecurity team and the penetration tester, where MITRE ATT&CK for ICS was discussed in detail and mapped to STRIDE threats. These discussions aimed to guide the penetration tester in planning the tests in a structured way. The following subsections describe the process of selecting potential tests based on MITRE ATT&CK for ICS and illustrate the procedure for mapping STRIDE categories to MITRE ATT&CK for ICS techniques.

4.3.1. Attack preconditions

As discussed in Khalil et al. (2023), to cause harm to a system, except for certain types of DoS attacks, an attacker must first gain access to the system, which is considered an attack precondition. Accordingly, three attack precondition stages were defined: *Initial Access, Discovery*, and *Lateral Movement*, as shown in Fig. 4. These stages are also defined as tactics in the MITRE ATT&CK for ICS. In a possible attack scenario,



Fig. 4. Attack precondition stages.

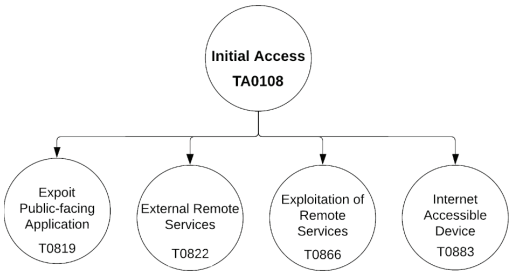


Fig. 5. MITRE ATT&CK for ICS *Initial Access* techniques targeted in the case study testing plan.

the tester gains initial access to the system using one of the *Initial Access* techniques (e.g., *exploit of public-facing application*) to attack the system from the internet. Then, the tester can proceed with system discovery using the *Discovery* techniques (for example, using *network sniffing*) and perform *Lateral Movement* (using *default credentials* or sniffed *valid accounts*).

In addition to the STRIDE categorization of MITRE ATT&CK for ICS techniques (see Section 4.3.2), possible precondition techniques that the tester might use to gain access to the system were also selected. Figs. 5, 6, 7 present only the selected precondition techniques for this case study. The full list of techniques related to each of the three chosen tactics can be found in the MITRE ATT&CK matrix for ICS.[5]

To gain initial access, it was agreed that the penetration tester would simulate an attacker attempting to compromise the system from the internet. Accordingly, four techniques were selected from the MITRE ATT&CK for ICS *Initial Access* tactic, as shown in Fig. 5. The *exploit public-facing application* and *exploitation of remote services* leverage software vulnerabilities to gain unauthorized access to the system. The *external remote services* technique targets services similar to VPN, potentially using valid accounts. While the *internet accessible device* technique aims to gain access to internet-exposed devices, intentionally or unintentionally, such as cellular modems, without the use of exploits.

The *remote services* technique was excluded as it targets internal services typically used by operators for remote management, making it more relevant to lateral movement within the context of this study. Additionally, MITRE ATT&CK for ICS *Initial Access* techniques requiring the tester's physical presence (e.g. *rogue master*), user interaction (e.g., *spearphishing attachment, drive-by compromise*), or those outside the case study scope (e.g., *supply chain compromise*) were also excluded from testing.

It was agreed that if remote access to the system is successfully gained from the Internet, the tester may proceed with network discovery to identify further vulnerabilities that could be exploited for lateral movement. However, if all attempts using the selected *Initial Access* techniques are unsuccessful, the tester may use the provided limited-access accounts to establish initial access and initiate the network discovery.

Four techniques were selected from the *Discovery* tactic, as shown in Fig. 6. The *network connection enumeration* might be used by the penetration tester to inspect the state of network connections and collect information about devices' roles. The *network sniffing* is a fundamental technique to collect information about the system. The technique might also be used to collect information about the used subnets and internal domains. The *remote system discovery* and the *remote system information discovery* techniques might be used to discover the system's remote services and collect information about these services, such as the configuration.

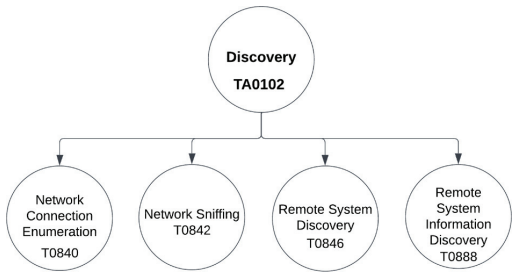


Fig. 6. MITRE ATT&CK for ICS *Discovery* techniques targeted in the case study testing plan.

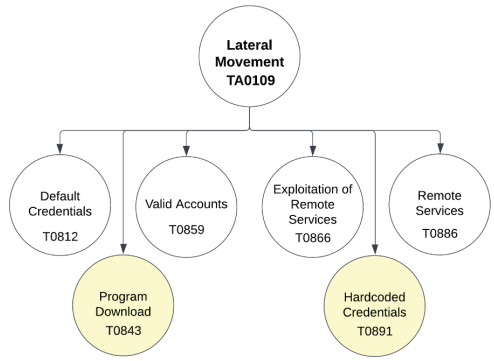


Fig. 7. MITRE ATT&CK for ICS *Lateral Movement* techniques targeted in the case study testing plan.

Following the *Discovery* stage, the tester might attempt to pivot between devices on the same network using *Lateral Movement* techniques. It was previously agreed that the testing scope encompasses lateral movement simulations across distinct network segments. The testing plan focused on four core techniques selected from the *Lateral Movement* tactic (as shown in Fig. 7), with an additional two techniques identified as potential candidates for further testing (highlighted in yellow).

Information gathered during discovery about devices and operating systems can be used to identify the manufacturer's default passwords, enabling the tester to employ the *default credentials* technique for lateral movement. Similarly, enumerated user or service accounts during discovery can be leveraged for the *valid accounts* technique. The *exploitation of remote services* technique, previously used for initial access, remains applicable for lateral movement within the system. Additionally, the *remote services* technique might be used to navigate the network laterally.

The *program download* technique was marked as a potential test due to its specific applicability to ICS environments. This technique often requires initial access to the OT network (Trust Zone 4) as a precondition. Similarly, the *hardcoded credentials* technique was identified for potential testing, depending on the available time frame. Exploiting these credentials might involve leveraging vulnerabilities to reveal them, which may require advanced system knowledge.

[5] https://attack.mitre.org/matrices/ics/

4.3.2. STRIDE-based categorization of MITRE ATT&CK for ICS techniques

Following the definition of attack preconditions, the next step focused on mapping the MITRE ATT&CK for ICS techniques to the STRIDE threat categories, to facilitate alignment between the threats identified during the threat modeling phase and the penetration testing plan. This approach helped to prioritize and assess the relevance of each categorized technique within the testing plan. However, this process also revealed that not all MITRE ATT&CK for ICS techniques represent threat types that can be meaningfully expressed using STRIDE. For example, several *Execution* techniques, such as those involving command-line and graphical user interfaces, describe how an unauthorized actor (or, in this case, the penetration tester) interacts with the system. These techniques could serve as attributes within one of the created STRIDE categorizations, but they do not fit neatly into the core STRIDE categories. Therefore, such techniques were not included in the categorization, which focused solely on those that could be clearly associated with one or more STRIDE threats.

Similarly, the *Command-and-Control* tactic in MITRE ATT&CK for ICS describes adversary communication mechanisms and operational behaviors rather than security threats in themselves. As such, these techniques do not map directly to *STRIDE* threat categories and are instead treated as enabling techniques that support the realization of STRIDE threats. In the same way, the *Impact* tactic in MITRE ATT&CK for ICS describes the consequences of successful attacks rather than the threats or adversary actions themselves. Therefore, all techniques included in the *Command-and-Control* and *Impact* tactics were excluded from this categorization as well.

The procedure used for categorization is illustrated in Fig. 8. The first author mapped the MITRE ATT&CK for ICS techniques using MITRE ATT&CK Navigator[6] by reviewing each technique description and annotating the corresponding STRIDE category (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, or Elevation of privilege) within the Navigator.

Some techniques were classified under multiple STRIDE categories. For example, the *adversary-in-the-middle* technique allows an attacker to intercept traffic between two nodes and collect sensitive information about the exchanged data. In such cases, the attacker could also modify packets, inject false data, or block traffic from reaching its intended destination. For this reason, the *adversary-in-the-middle* technique was categorized under information disclosure, tampering, spoofing, and denial of service. The *rogue master* technique was also categorized under spoofing and tampering, as it refers to a malicious controller that impersonates a legitimate controller to issue control commands and may also tamper with them.

The first author then initiated a meeting between the cybersecurity team and the penetration tester to discuss the proposed categorization and agree on the techniques to be included in the test plan. For each technique, a decision was made regarding its inclusion or exclusion, and the feasibility of executing each selected technique was further examined.

It should be noted that the initial categorization was performed using MITRE ATT&CK for ICS v13, while the information was later updated to align with v18, which was the latest version available at the time of writing. MITRE ATT&CK for ICS v18 includes 83 techniques, of which 46 were directly mapped to STRIDE threat categories. Techniques that were applicable to more than one STRIDE threat type were counted only once in this total. When additionally considering techniques relevant to the defined attack preconditions (discussed in Section 4.3.1), including all techniques from the *Initial Access*, *Discovery*, and *Lateral Movement* tactics, and excluding duplicates already captured within the STRIDE categorization, the total number of potential techniques considered for ICS penetration test planning amounted to 64 out of the 83 techniques.

[6] https://mitre-attack.github.io/attack-navigator/

Some techniques require significantly more time and effort than others and may also depend on certain preconditions that must be met. For example, executing an attack based on the *unauthorized command message* technique would first require gaining initial access to the OT network (Trust Zone 4). Without this access, the penetration tester would be unable to perform the corresponding test. Such advanced techniques were marked for potential inclusion, depending on the availability of time and fulfillment of necessary preconditions. The remainder of this section details the proposed STRIDE categorization of MITRE ATT&CK for ICS techniques.

Spoofing: Eight techniques were categorized as spoofing, as shown in Fig. 9. The spoofing tests included four core techniques (*default credentials, adversary-in-the-middle, valid accounts, execution through API*) to be executed during the exploitation stage. Three techniques were identified as potential tests, to be performed if time and access permitted: *masquerading* and *hardcoded credentials* (conditional on time), and *unauthorized command message* (conditional on gaining initial access to the OT network, Trust Zone 4, which was not provided to the tester). The *rogue master* technique was excluded from the testing scope, as the assessment focused on attacks that can be launched remotely, without physical access to the system. Finally, to prioritize testing of the in-house control software, techniques relevant to that software, similar to *valid accounts* and *execution through API*, were given precedence.

It is to be noted that the *execution through API*, one of the techniques under the *Execution* tactic, was mapped to spoofing only in scenarios where API calls could be issued without proper authentication or where an adversary could impersonate a trusted system component, thereby violating identity guarantees.

During the threat modeling phase, the most critical threats were spoofing threats. Using the proposed spoofing categorization for MITRE ATT&CK for ICS techniques, the tester may simulate an attacker's behaviors. For instance, the tester may combine the *adversary-in-the-middle* technique with the use of *valid accounts* to mimic a threat agent who gains unauthorized access and impersonates a Microgrid administrator, potentially disrupting critical operations. Additionally, the tester can evaluate the security of API endpoints by attempting to execute commands and assessing whether authentication is required. Techniques such as using devices' default passwords (i.e., *default credentials*) or brute-forcing accounts (i.e., *valid accounts*) can also be employed to identify potentially weak authentication mechanisms.

Tampering: Eighteen techniques were categorized as tampering, as detailed in Fig. 10. However, only the *adversary-in-the-middle* was explicitly included in the tampering testing plan. The categorization of tampering techniques sparked interesting discussions. Techniques similar to *block reporting message* might seem more suited for DoS, since it disrupts the monitoring services. However, it was also included in the tampering category. Consider a scenario where a system configuration always displays the last received monitoring data without raising alarms for missing reports. In such a case, blocking reporting messages during an attack could be considered tampering with the system's ability to report accurate information. The same logic applies to *block command message*. For example, if an attacker blocks a message containing a new setpoint value (a desired operating condition), the system will not receive the update and will remain on the previous setting. This effectively tampers with the system configuration established by the administrator.

Other tampering techniques were de-prioritized as it was considered that the successful execution of the *adversary-in-the-middle* technique could serve as a proof of concept for techniques such as *block reporting message* and *block command message*, if the testing timeframe did not allow those tests to be performed. The *system firmware* and *module firmware* were excluded as the embedded device attack surface, including firmware, lay outside the scope of this plan. The *Rootkit, rogue master, exploitation for evasion* techniques were also excluded.

It should be noted that the *modify controller tasking* technique, part of the *Execution* tactic, was introduced in a later update of the categorization and was therefore incorporated into Fig. 10. This technique

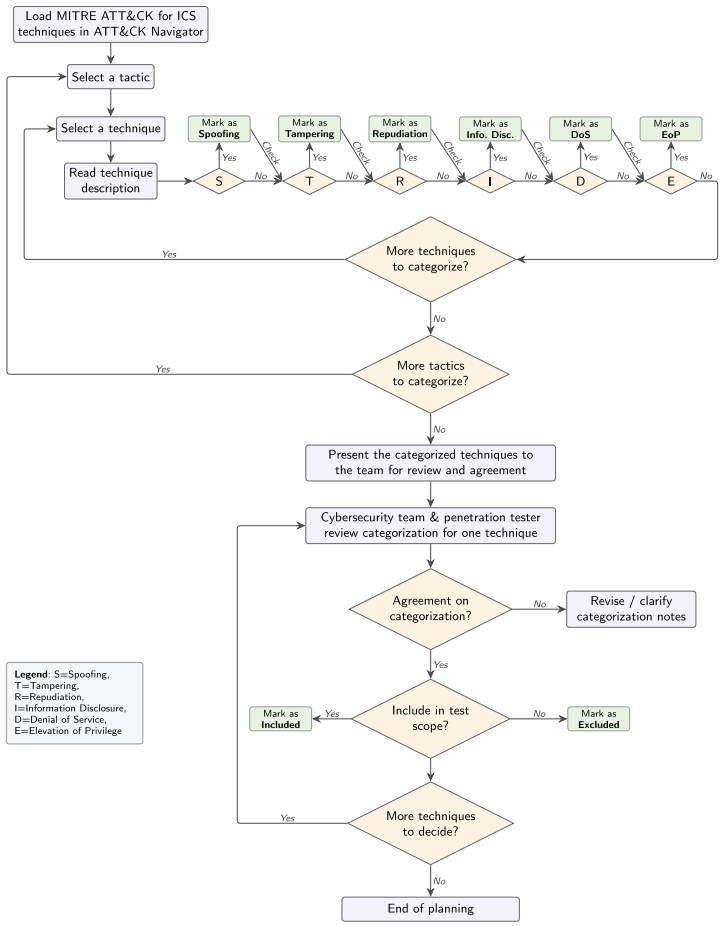


Fig. 8. Description of the process used to categorize MITRE ATT&CK for ICS techniques into STRIDE threat categories.

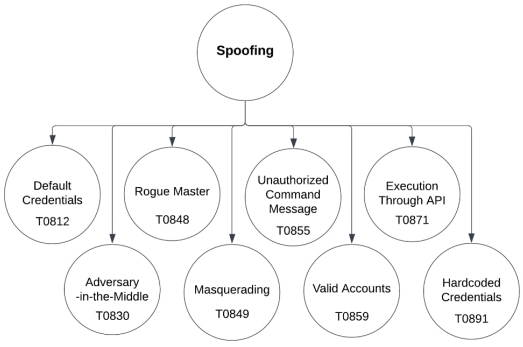


Fig. 9. Proposed categorization of Spoofing techniques for MITRE ATT&CK for ICS.

represents tampering due to the unauthorized modification of program task associations and execution properties, which directly alter controller behavior. As the penetration testing plan was defined prior to this update, *modify controller tasking* was not considered during the testing plan.

Repudiation: Six techniques were categorized as repudiation techniques, as shown in Fig. 11. However, it was decided to exclude all the techniques related to repudiation from the testing plan as the losses and consequences of the repudiation threats were minimal compared to other threats. The repudiation techniques mainly aim to hide the attacker's traces, which was set out of the scope of the testing plan, as the main goal was to simulate attacks that might cause a critical impact on the system.

Information Disclosure: Eleven techniques were categorized as information disclosure techniques, as shown in Fig. 12. Notably, the eleven techniques categorized under Information Disclosure all fall within the *Collection* tactic of MITRE ATT&CK for ICS, indicating a consistent alignment between the two classifications. During the threat modeling, information disclosure threats were classified as low-priority due to their limited impact on system functionality. However, four information disclosure techniques were still included in the testing plan: *adversary-in-the-middle*, *data from information repositories*, *data from local system*, and *screen capture*. While in real attack scenarios, the *screen capture* technique is mostly combined with malware infection or other automation techniques, it is a vital tool in a penetration testing

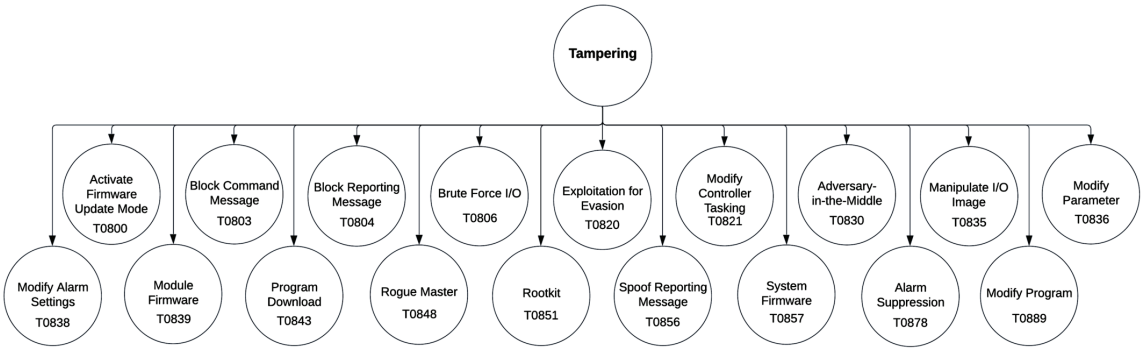


Fig. 10. Proposed categorization of Tampering techniques for MITRE ATT&CK for ICS.

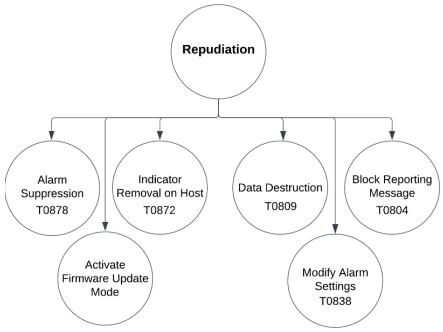


Fig. 11. Proposed categorization of Repudiation techniques for MITRE ATT&CK for ICS.

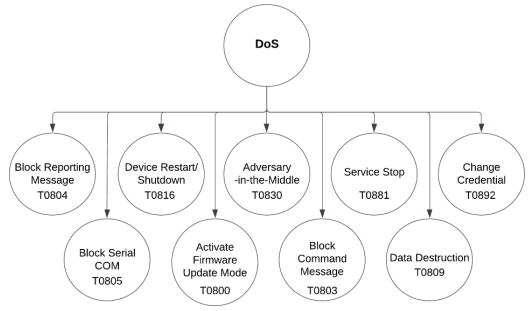


Fig. 13. Proposed categorization of Denial of Service techniques for MITRE ATT&CK for ICS.

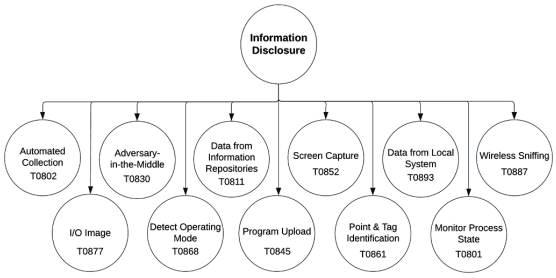


Fig. 12. Proposed categorization of Information Disclosure techniques for MITRE ATT&CK for ICS.

exercise as well. It documents the tester's accomplishments and the types of information exposed during the simulated breach. The primary goal in incorporating these four techniques was to understand the potential information an attacker could access and how the collected information may help in an attack progression.

Five information disclosure techniques, however, were de-prioritized: *I/O image, detect operation mode, program upload, point & tag identification*, and *monitor process state*. These techniques require gaining access to the OT network (Trust Zone 4), a prerequisite not granted for this testing phase. The *automated collection* technique was excluded due to the potential time commitment required for scripting and automation tool development, and given the low priority of the information disclosure threats. *Wireless sniffing* was also excluded, as there is no wireless network in the target system.

Denial of Service: Nine techniques were categorized as DoS, as illustrated in Fig. 13. Since the target system was not yet operational during testing, the *adversary-in-the-middle* technique was selected for execution to evaluate the system's resilience to denial-of-service conditions using a controlled and non-destructive testing approach. Successfully performing this technique was considered sufficient to demonstrate the feasibility of other related DoS techniques that rely on intercepting or blocking communication. For example, if the tester successfully spoofs the default gateway's MAC address using a device that cannot process such traffic, this should result in a denial of service. The use of the *adversary-in-the-middle* technique in such a case proves that other techniques similar to *block reporting message* and *block command message* are also possible. The *block serial com* technique was excluded from testing. The remaining techniques were retained as potential testing candidates if sufficient time was available and the necessary preconditions were met during the assessment, provided that they could be executed using controlled and non-destructive testing approaches.

Elevation of Privilege: Finally, eight techniques were categorized as elevation of privilege techniques, as illustrated in Fig. 14 , reflecting the updated categorization aligned with MITRE ATT&CK for ICS v18. The *exploitation for privilege escalation* and *valid accounts* techniques were included in the penetration testing plan, with the potential to add the *hooking* and *change of operation mode* techniques if the tester gained access to the OT network. The *system firmware* technique was excluded, as discussed in the tampering categorization.

It should be noted that three techniques shown in Fig. 14, *autorun image*, *system binary proxy execution*, and *native API*, were incorporated during a later update of the categorization when aligning the study with MITRE ATT&CK for ICS v18. The first two techniques were introduced in later ATT&CK versions, while *native API* already existed in v13 but was not previously categorized under Elevation of Privilege. As these

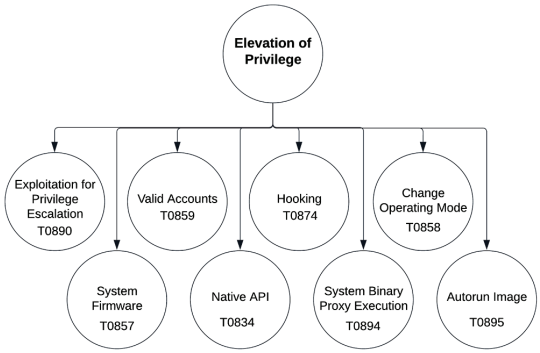


Fig. 14. Proposed categorization of Elevation of Privilege techniques for MITRE ATT&CK for ICS.



Fig. 15. Comparison of the severity ratings assigned to reported vulnerabilities using CVSS v3.1 and CVSS v4.0.

techniques were not part of the original penetration test planning, they were not included in the executed testing scope. They are mapped to Elevation of Privilege only when they lead to unauthorized privilege escalation through the abuse of trusted system mechanisms.

The planning stage aimed to create a high-level test plan that considered the prioritized system threats. When the tests are performed by a third party, such a plan should guide the testers through the exploitation stage, while giving them the freedom to choose the testing tools and procedures. However, for tests performed internally, the plan could be further extended to include more detailed testing procedures, with commands to be used and tools.

4.4. Exploitation

The main goal of this case study is to discuss the penetration testing process, with a primary focus on the planning. The third party fully conducted the exploitation stage without the involvement of the cybersecurity team. Furthermore, the tests were performed on a real-world, operational system. For these reasons and considering the confidentiality of the exploitation stage results, they are excluded from the scope of this paper.

4.5. Vulnerability ranking

Twelve software vulnerabilities were reported by the penetration tester. Notably, not all of the reported vulnerabilities were directly related to traditional OT devices, although they could still have severe impacts on the system's safety and availability. As previously described, the primary testing focus was the in-house software platform responsible for microgrid management, which, although implemented as software, operates as a critical OT component within the system architecture. Consequently, several findings reflect software and web-exposed weaknesses that could still affect operational safety and continuity. These included, for example, weak credentials, SQL injection, unencrypted communications, and authentication-related misconfigurations identified on management and communication interfaces.

The tester categorized the reported vulnerabilities using the CVSS v3.1 calculator[7] based on the base metric. The severities of these vulnerabilities were rated as three critical, two high, three medium, and four low. As CVSS v4.0 incorporates safety and resiliency aspects, the first author later re-categorized the same vulnerabilities using the CVSS v4.0 calculator,[8] while considering the supplemental, environmental, and threat metrics.[9] This resulted in a revised severity distribution of four critical, two high, two medium, and four low vulnerabilities, as shown in Fig. 15. Overall, the observed score shifts were primarily driven by the incorporation of subsequent system impact, system security requirements, and safety-related consequences in CVSS v4.0. Severity differences between the two versions were minimal for vulnerabilities with limited operational implications.

Table 3 provides a detailed comparison of selected vulnerabilities to illustrate how contextual factors in CVSS v4.0 influence severity assessment. Vulnerability number one represents a grouped finding in which multiple technical weaknesses were reported collectively, resulting in high confidentiality, integrity, and availability impact values under CVSS v3.1. Accordingly, it received a base score of 8.8 (High). When re-evaluated using the CVSS v4.0 base metrics, the score remained comparable at 8.7 (High), since no impact on subsequent systems was identified.

However, when applying the CVSS v4.0 environmental metrics, the system context identified during the requirement selection phase was explicitly incorporated. The Confidentiality Requirement (CR) had been determined as Low during the threat-modeling exercise, while the Integrity Requirement (IR) and Availability Requirement (AR) were set to High to reflect operational priorities. As this grouped vulnerability predominantly affected confidentiality aspects and did not introduce safety-relevant consequences, the adjusted CVSS v4.0 score slightly decreased to 8.6 (High). This example illustrates how CVSS v4.0 enables contextual refinement of severity based on system security requirements rather than relying solely on generic technical impact assumptions.

A key distinction in CVSS v4.0 is the explicit modeling of subsequent system impact and safety-related consequences. Vulnerability number two in Table 3 corresponds to a demonstrated denial-of-service attack against a system component. During the threat-modeling phase, denial of service on this component had already been associated with operational consequences such as interruption of control functions, limited energy availability, insufficient power availability, and potential loss of energy sold. Under CVSS v3.1, the vulnerability received a base score of 7.4 (High) primarily due to the availability impact on the affected component. When evaluated using CVSS v4.0 base metrics, Availability impact on both the Vulnerable System (VA) and the Subsequent System (SA) was set to High to reflect the modeled downstream operational effects, resulting in an increased base score of 8.3 (High). Upon incorporation of environmental and safety-related metrics, including recognition

[7] https://www.first.org/cvss/calculator/3.1
[8] https://www.first.org/cvss/calculator/4.0

[9] This action was performed during the report Analysis, remediation, and retesting stage (see Section 4.8), although the results are presented under the vulnerability ranking section to facilitate readability and comparison.

Table 3
Severity comparison between the CVSS v3.1 base score and the CVSS v4.0 base and adjusted scores for a sample of reported vulnerabilities.

| Vulnerability number | CVSS v3.1 Base Score Calculator | | | | | | | | | CVSS v4.0 Calculator | | | | | | | | |

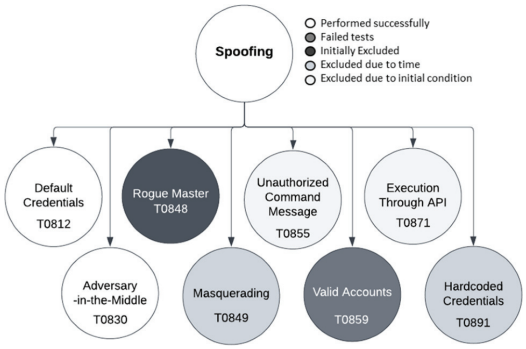


Fig. 16. An illustrative example of the color-coded approach using MITRE ATT&CK for ICS categorized spoofing techniques.

was submitted; however, its results are presented in the vulnerability ranking section (Section 4.5) to facilitate readability and enable direct comparison with the CVSS v3.1 scores. As part of this reassessment, the cybersecurity team retested selected vulnerabilities and communicated with stakeholders to confirm that all testing accounts had been deleted and that the proposed remediation had been implemented on the system.

As the penetration testing report identified six critical and high vulnerabilities, a subsequent analysis was conducted to validate the research hypothesis by examining the reported vulnerabilities and their required mitigation strategies. This analysis compared the proposed mitigation strategies for each reported vulnerability against the security requirements established during the design phase. The goal was to determine whether the proposed mitigation strategies were already addressed by the security requirements and to identify any potential requirement gaps. Additionally, the analysis assessed whether the security requirements had been properly considered during the design phase, or whether they were overlooked or not implemented correctly.

As an illustrative example of how the reported vulnerabilities were analyzed and traced, one of the identified findings involved an SQL injection flaw. Such attacks may be categorized as either *Information Disclosure* or *Tampering*, depending on their resulting impact on the system. Within the *MITRE ATT&CK Enterprise* matrix, SQL injection is explicitly mapped to *Exploit Public-Facing Application (T1190)*. In contrast, the technique with the same name in the *MITRE ATT&CK for ICS* matrix, *Exploit Public-Facing Application (T0819)*, does not explicitly reference SQL injection. This distinction reflects the higher level of abstraction adopted in the ICS matrix, which emphasizes operational effects rather than specific IT exploitation mechanisms. It also highlights the necessity of extending the categorization process to incorporate relevant techniques from the Enterprise (and, where applicable, Mobile) matrices to ensure more comprehensive technique coverage, given that ICS environments include both IT and OT systems.

In the reported vulnerability, the tester successfully exploited the SQL injection vulnerability to extract valid user account credentials from the compromised system component. This action constitutes unauthorized access to stored credential data and may align with the *Data from Local System (T0893)* technique in the ICS matrix, which was categorized under information disclosure (See Fig. 12). The extracted credentials were later used to authenticate as a legitimate user, which aligns with the *Valid Accounts (T0859)* technique. The tester then performed a spoofing attack on the compromised component. This attack demonstrates a threat scenario that was discussed during the planning phase.

The potential threat consequences associated with spoofing an administrative account of this component were already identified during the threat elicitation conducted during the design phase (Khalil et al., 2023). By successfully masquerading as the component administrator, an attacker would obtain authorized access to privileged control functions within the system. This access could enable the modification of operational parameters (e.g., setpoints, protection thresholds, load balancing configurations), alteration of grid-connection or disconnection commands, manipulation of energy dispatch logic, disabling of alarms, or interference with monitoring and visibility functions.

the relevant threat consequences that were identified during the design phase, included overload or exceeding operational limits, interruption of energy transfer, Interrupting/manipulating the control, inability to provide ancillary services, misleading safety operations, limited energy availability, misleading connection/disconnection operations, violation of power quality, insufficient power availability, loss of energy sold, insufficient visibility of inverters, and leaking information for further attacks. Although the vulnerability originates at the software level, the resulting attack chain demonstrates how compromise of administrative credentials may propagate across control logic and operational processes, ultimately leading to severe operational and safety consequences in an ICS environment.

During the threat-modeling phase, the selected security requirements included IEC 62443-4-2 component requirement (CR) 3.5, *Input Validation*. CR 3.5 requires that components validate the syntax, length, and content of data received via external interfaces. Proper implementation of this requirement is expected to mitigate injection-based attacks, including SQL injection. Therefore, the existence of the reported vulnerability indicates insufficient enforcement or improper implementation of input validation controls.

Additional requirements, such as CR 1.1, *Human User Identification and Authentication*, including Requirement Enhancement (RE) 1, *Unique identification and authentication*, and CR 1.7, *Strength of Password-Based Authentication*, including RE (1), *Password generation and lifetime restrictions for human users*, were also specified during the design phase. However, the exploitation revealed that the extracted credentials corresponded to a weak password configuration, indicating inadequate enforcement of password policy controls.

While the primary entry point of the attack was the input validation weakness, the presence of weak credential management significantly increased the exploitability and impact of the vulnerability. Accordingly, the root cause of the attack reflects both insufficient input validation and ineffective enforcement of authentication hardening measures, demonstrating a breakdown in defense-in-depth implementation rather than a deficiency in requirement selection.

Another reported category was vulnerable software. Unlike authentication flaws, this finding does not correspond to a specific STRIDE threat category; rather, it represents an attack precondition that may enable subsequent exploitation techniques, such as exploitation of remote services or public-facing applications within the MITRE ATT&CK for ICS matrix. During the requirement selection phase, IEC 62443-4-2 CR 3.10 *Support for Updates* was included to ensure that system components provide the technical capability to perform upgrades and updates. However, IEC 62443-4-2 does not define requirements for structured patch management processes or mandatory update cycles; such governance-level controls are addressed separately in IEC TR 62443-2-3 (IEC, 2015), which provides guidance on patch management processes in industrial automation and control system environments. Therefore, the presence of vulnerable software reflects deficiencies in lifecycle and update governance practices rather than an omission in the selected component-level security requirements.

The qualitative review revealed that mitigation techniques for nine of the twelve reported vulnerabilities had already been defined through the selected security requirements. In these cases, proper implementation and enforcement of the identified controls would have prevented or significantly limited the respective attacks. The vulnerable software finding reflected lifecycle and patch governance deficiencies beyond the scope of the selected component-level requirements. The remaining two low-severity vulnerabilities were not explicitly prioritized during the threat-modeling phase due to their limited operational impact;

however, their identification still required corrective action by the system owner, reflecting one of the core objectives of performing system-level penetration testing. These results align with the research hypothesis that penetration testing can serve as a validation mechanism for design-phase threat modeling. The findings further underscore the importance of conducting security requirements testing and threat mitigation testing before initiating penetration testing. Such preparatory work increases the likelihood of identifying higher-impact vulnerabilities during testing, particularly when time constraints limit the depth of third-party assessments.

Comprehensive reporting and subsequent analysis of the penetration testing results provided valuable insights into the system's security posture. The report, together with the review meetings, ensured a clear understanding of the performed tests, discovered vulnerabilities, and their corresponding mitigation strategies. Ultimately, this process validated the relevance of several security requirements while also identifying critical gaps in their implementation. These results further demonstrate the effectiveness of threat modeling-driven penetration testing as a structured approach for the security verification and validation testing of an ICS.

5. Discussion

Threat modeling is an analytical design-phase process intended to identify critical threats and propose mitigation strategies. However, the threat modeling process by itself does not guarantee that the resulting system will achieve a minimum acceptable level of security. Even when security requirements are derived during threat modeling and communicated to the development team, practical constraints such as budget limitations, time pressure, or implementation challenges may lead to incomplete or incorrect implementation of these requirements.

The IEC 62443 standard outlines four distinct testing processes for security verification and validation testing of a newly designed system component: security requirements testing, threat mitigation testing, vulnerability testing, and penetration testing , where each process is executed independently. In such a setting, security requirements testing and threat mitigation testing should precede penetration testing, along with a more extensive vulnerability assessment beyond known vulnerabilities. Penetration testing would then focus on assessing the residual attack surface of the implemented system rather than validating individual security requirements.

However, in small and medium-sized ICS projects with limited resources, such as the project examined in this case study, where IEC 62443 certification was not targeted, stakeholders may decide to combine these processes to streamline the security assessment. Due to resource constraints, this case study prioritized penetration testing over security requirements testing and threat mitigation testing. Nevertheless, the security engineering activities were conducted with the intent of aligning with IEC 62443 principles, despite the constraints of a real-world project.

Several penetration testing methodologies integrate a threat modeling phase to support penetration test planning, hereafter referred to as threat modeling-based methodologies. Methodologies such as PatrIoT (Süren et al., 2023) and PTES (Penetration Testing Execution Standard, 2026) extend into the exploitation stage by providing structured guidance on how to derive attack paths and execute them. Such approaches focus on exploitation, as their primary objective is to guide the tester in carrying out technical attack procedures.

In contrast, the objective of the threat modeling-driven penetration testing methodology proposed in this study is not to prescribe detailed exploitation steps but to bridge the system design phase with penetration testing activities. The BRIDGE methodology focuses on guiding penetration test planning based on the identified critical threats, and may also be used, when IEC certification is not aimed, to verify that security requirements have been correctly implemented and to validate that threats identified for mitigation have been effectively addressed. In this sense, the proposed approach is also verification- and validation-driven. Importantly, methodologies that provide exploitation procedures can still complement the BRIDGE methodology in the exploitation stage, since BRIDGE does not prescribe how exploitation should be technically performed, but instead ensures that penetration testing is systematically aligned with the threats identified during threat modeling and the corresponding security requirements defined during system design. The key distinctions between threat modeling-based methodologies and the BRIDGE methodology are summarized in Table 4.

Table 4
Comparison of BRIDGE, a threat modeling-driven penetration testing methodology, with existing threat modeling-based penetration testing methodologies.

Aspect	PTES	PatrIoT	BRIDGE
Integration of design-phase threat modeling	✗	✗	✓
Threat modeling supports penetration test planning	✓	✓	✓
IoT-oriented	✗	✓	✗
ICS-oriented	✗	✗	✓
Security requirements verification and validation	✗	✗	✓
Provides technical procedures	✓	✓	•

Symbols: ✓ = present; • = partial or medium-level coverage; ✗ = not provided.

Documenting security requirements derived from prioritized threats in alignment with IEC 62443 is part of the first phase of the BRIDGE methodology and was addressed in previous work on the same microgrid project (Khalil et al., 2023, Section 5.9). As the present study focuses on penetration test planning, the full set of security requirements was not required for the planning and execution of the testing activities and was therefore not shared with the penetration tester. Instead, as described in Section 4.1, the tester was provided with the updated system model, the data flow diagram, and prioritized threats together with their anticipated consequences. This reflects common industrial practice, where the validation of individual security requirements remains outside the scope of penetration testing. Following test execution, the penetration testing findings were retrospectively analyzed by the internal cybersecurity team against the design-phase security requirements (Section 4.8) to assess whether these requirements had been effectively addressed.

Even in a fully compliant certification process, penetration testers would not require direct access to the security requirements, as these are addressed and verified through earlier, independent testing activities. While this separation of testing processes could not be fully realized in the present resource-constrained project, the documentation produced across the BRIDGE methodology preserves traceability and would support a more extensive, certification-driven testing process if pursued in the future.

The results of this case study highlight a potential need for prioritizing security requirements alongside threat prioritization during the design phase. While prioritizing threats is crucial, it can result in an extensive list of security requirements, especially when multiple requirements are assigned to each threat. This may obscure which requirements are truly critical. To address this, prioritizing security requirements themselves can help ensure that, when multiple security requirements are provided to a single threat, the most critical requirements receive appropriate attention throughout the design process. Additionally, negotiating and following up on the implementation of security requirements with system owners helps ensure that those associated with critical and high-priority threats are properly implemented.

Accordingly, the results of this study suggest that security requirements testing and threat mitigation testing should be performed before conducting penetration testing whenever possible. Running these two processes before penetration testing may reveal critical vulnerabilities that were supposed to be mitigated by design, allowing system owners to address these vulnerabilities before the penetration testing. This, in turn, would enable penetration testers to focus on more advanced

attack scenarios. In such cases, the testers may discover critical attack vectors that were not considered or were low-priority during the threat modeling phase. These vulnerabilities may result from unforeseen system interactions, configuration weaknesses, or zero-day vulnerabilities. Hence, the penetration testing results can further validate or refine the threat priorities identified during threat modeling. If a seemingly low-priority threat is shown to have a critical impact during testing, it can be addressed with greater urgency.

In practice, engaging a third party for penetration testing offers valuable expertise, but it also has limitations. While the cybersecurity team may strive to guide the testing plan to encompass both IT and OT security testing, a key challenge lies in achieving the right balance between IT and OT testing activities. Maintaining high-level involvement during the planning stage is recommended for effective collaboration with external testers. However, outsourcing the testing to a thrid party limits the ability of the cybersecurity team to define a detailed testing plan. High-level planning with the tester can ensure that the critical threats are tested. However, the tester may still prioritize certain types of tests over others. Overlooking OT-specific tests may potentially result in a bias towards IT vulnerabilities compared to those specific to the OT environment. Nevertheless, this case study demonstrates that IT vulnerabilities can still have a significant impact on ICS safety.

Consistent with these practical constraints, the vulnerabilities revealed in this microgrid case study were primarily software-related in nature. The identified vulnerabilities correspond to common software-level vulnerability classes typically assessed in IT penetration testing, such as authentication, cryptographic, and application-level weaknesses, even when these vulnerabilities are present on OT components. The predominance of such vulnerabilities can be attributed to the focus on remote, network-based testing and the practical constraints of limited testing time. Nevertheless, the analysis showed that these software-level vulnerabilities could still have critical impacts on the OT system, as they may expose control-relevant components and lead to safety and availability consequences.

The case study highlights the advantages of CVSS v4.0 for ICS vulnerability scoring. Compared to CVSS v3.1, CVSS v4.0 offers a more effective assessment by incorporating system security requirements (confidentiality, integrity, and availability) and the vulnerability's impact on the safety of subsequent systems. These factors lead to a more relevant and accurate CVSS v4.0 score for ICS environments.

Due to the inherent complexities of a real-world case study, the research encountered challenges that redirected the investigation in unforeseen directions. While many CPS threat modeling methods exist, STRIDE was selected for the threat modeling phase as it offers a balance between simplicity and clarity. STRIDE focuses on basic cybersecurity concepts, which makes it easy to assimilate for stakeholders with limited or no cybersecurity background. Notably, during the early design phase, a high-level threat modeling method was required to support the choice of security requirements. A more detailed taxonomy could have elicited a broader range of threats, which could also increase the complexity and time commitment of the threat modeling exercise, potentially hindering its use in future projects. In contrast, penetration test planning requires a middle-level taxonomy that provides sufficient detail about ICS attack types while remaining manageable for high-level plan creation, especially since the testing was conducted by a third party. These practical constraints justify the need to map MITRE ATT&CK for ICS techniques to STRIDE threats.

This paper focuses on linking design-phase threat modeling outcomes with subsequent security testing activities to improve traceability across the secure development life cycle. Mapping STRIDE threat categories to MITRE ATT&CK for ICS techniques enables design-phase threats to be systematically carried forward into penetration test planning. Both STRIDE and MITRE ATT&CK are qualitative, expert-driven frameworks; therefore, the proposed mapping is intentionally interpretive rather than rule-based. Certain MITRE ATT&CK for ICS techniques are associated with multiple STRIDE categories when they contribute to different threat objectives, reflecting real-world attack behavior rather than enforcing a strict one-to-one mapping. By explicitly documenting these overlaps, exclusions, and cases where the mapping is based on explicit justification, the approach improves transparency and supports reproducibility.

The case study further revealed that threat modeling-driven penetration testing requires a reporting structure that differs from traditional vulnerability-class-based formats. Unlike conventional reports, such as those structured around OWASP testing classes, the BRIDGE methodology requires findings to be explicitly mapped to elicited threat scenarios, associated system components, and the relevant techniques used during the exploitation phase, while documenting the attack results (as shown in Fig. 16). Moreover, as ICS environments may involve web, IoT, and mobile interfaces, testing techniques may span multiple frameworks beyond a single guide. Accordingly, future applications of BRIDGE should formally define a reporting template that supports cross-framework technique integration and preserves traceability between design-phase threat modeling outputs, penetration test planning decisions, and exploitation results.

The inherent confidentiality of ICS penetration testing reports, due to the sensitive nature of system design and vulnerabilities, poses a challenge for academic publication. In this case study, the penetration testing procedures are discussed without disclosing specific system details, such as network protocols, vendors, or device models. Information about the system components targeted by the tests, successful and failed tests, and detailed descriptions of the identified vulnerabilities are also omitted. Instead, the STRIDE-based categorization of MITRE ATT&CK for ICS techniques is presented, highlighting how this categorization can guide the tester. Additionally, general information about some vulnerability categories (e.g., software, DoS) is provided to support reader comprehension while preserving confidentiality. This approach protects the operational system from potential targeted attacks based on the level of information provided in this paper.

Taken together, the findings of this case study highlight both methodological and practical contributions for threat modeling-driven penetration testing in ICS environments. By balancing confidentiality with knowledge sharing, the BRIDGE methodology provides a valuable reference for future threat modeling-driven penetration testing projects of newly deployed ICS. In addition, there is also a notable scarcity of resources that comprehensively describe the real-world execution of ICS threat modeling-based penetration testing, including what types of information should be discussed at each stage and which decisions should be made. This paper aims to contribute to this knowledge base by documenting these aspects alongside the case study results. As a result, the findings not only validate the methodology but also offer practical guidance for practitioners facing similar constraints in real ICS deployments.

6. Threats to validity

This section outlines the limitations of the proposed methodology and the case study, and identifies potential threats to the validity and generalization of the reported results.

The proposed methodology is intended for planning the initial penetration test of a newly deployed ICS rather than serving as a general-purpose penetration testing framework. Certain methodological choices made during the case study, such as the mapping of MITRE ATT&CK techniques to STRIDE categories, may be applicable beyond the evaluated system. However, it is important to keep in mind that the categorization was developed specifically in the context of penetration testing and was informed by expert discussions within the project team.

Some of the technique categorizations may be open to debate, and in future repetitions of the process, the team might decide to revise certain classifications. For this reason, the categorization process and its underlying rationale are shared to facilitate reproducibility, as further categorizations for other MITRE matrices may also be required for penetration testing. This is particularly relevant because an ICS attack can

combine techniques originating from different matrices, including ICS, Enterprise, and Mobile matrices.

The methodology focuses on aligning threat modeling outputs with test selection and prioritization; it does not aim to provide a technical, step-by-step exploitation guide for the execution stage of penetration testing. For operational exploitation and post-exploitation procedures, the methodology is designed to be combined with existing technical guides and best-practice resources. Thus, effectiveness claims are limited to planning, prioritization, and traceability rather than exploitation success rates.

This study relies on a single real-world case study and a single penetration testing campaign, which limits the ability to assess robustness, repeatability, and comparative advantages over alternative penetration test planning approaches. The study aims to demonstrate the feasibility and practical value of systematically linking design-phase threat modeling outputs to penetration test planning in an ICS context. The case study, therefore, serves as an exploratory validation, illustrating how the BRIDGE methodology can be applied under realistic industrial constraints. The methodology and its underlying decision rationale are thoroughly documented to ensure transparency and reproducibility. Further validation through additional case studies across different ICS domains and repeated testing campaigns will be required to assess robustness and broader applicability. Nevertheless, the presented case study provides an initial empirical demonstration of how threat modeling outcomes can be systematically incorporated into ICS penetration test planning.

7. Conclusion

This research highlights the value of integrating design-phase threat modeling outcomes into ICS penetration testing plans. To demonstrate how this integration can be achieved, the BRIDGE methodology, a nine-stage threat modeling-driven ICS penetration testing methodology, was proposed and its practical application was demonstrated through a real-world case study. A STRIDE-based categorization of MITRE ATT&CK for ICS techniques was also introduced to support structured test planning. The detected system vulnerability scores were calculated using both CVSS v3.1 and CVSS v4.0. The results indicate that CVSS v4.0 is more effective in capturing the unique risk characteristics of ICS vulnerabilities, particularly those with potential safety or physical consequences. This study contributes a structured approach to ICS penetration testing that improves planning and helps reduce the likelihood of missing critical vulnerabilities. While the findings are promising, further research is needed to develop a comprehensive library of ICS-specific testing procedures. By streamlining test planning and aligning the testing phase with design-phase insights, the proposed methodology supports both practitioners and researchers in enhancing the security and resilience of industrial control systems.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used Google's Gemini and OpenAI's ChatGPT to assist with grammar review and language refinement. No content was generated by these tools; the authors solely authored the content and used AI tools for editorial suggestions. All edits were reviewed and approved by the authors, who take full responsibility for the final version of the manuscript.

CRediT authorship contribution statement

Shaymaa Mamdouh Khalil: Conceptualization, Methodology, Investigation, Data curation, Writing - original draft, Writing - review & editing, Visualization; **Hayretdin Bahsi:** Supervision, Conceptualization, Methodology, Writing - review & editing; **Tarmo Korõtko:** Funding acquisition.

Data availability

The data that has been used is confidential.

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.

Acknowledgments

The authors gratefully acknowledge *Alvaro Wim Schuller Fernandez* for conducting the penetration testing activities described in this paper, and for the valuable contributions throughout the testing lifecycle. This work was supported by the Estonian Ministry of Education and Research and the European Regional Development Fund (grant 2014-2020.4.01.20-0289) and by the project *"Increasing the knowledge intensity of Ida-Viru entrepreneurship"* co-funded by the European Union (2021-2027.6.01.23-0034).

Supplementary material

Supplementary material associated with this article can be found in the online version at 10.1016/j.cose.2026.104906.

References

Alexander, O., Belisle, M., Steele, J., 2020. Mitre att&ck for industrial control systems: design and philosophy. The MITRE Corporation: Bedford, MA, USA 29, 21–85.

Bahsi, H., Dola, H.O., Khalil, S.M., Korõtko, T., 2022. A cyber attack taxonomy for microgrid systems. In: 2022 17th Annual System of Systems Engineering Conference (SOSE), pp. 324–331. https://doi.org/10.1109/SOSE55472.2022.9812642

Choi, S., Choi, J., Yun, J.H., Min, B.G., Kim, H.C., 2020. Expansion of ICS testbed for security validation based on MITRE ATT&CK techniques. CSET 2020 - 13th USENIX Workshop on Cyber Security Experimentation and Test, co-located with USENIX Security 2020 .

Commission, I.E., 2019. IEC 62443-4-2 : Security for industrial automation and control systems . Part 4-2 : technical security requirements for IACS components .

Commission, I. E., 2018. Security for industrial automation and control systems - Part 4-1: secure product development lifecycle requirements.

Commission, I.E., 2022. IEC 62443-2-1 Industrial communication networks. Network and system security. Part 2-1: establishing an industrial automation and control system security program .

Da Silva, M., Puys, M., Thevenon, P.H., Mocanu, S., Nkawa, N., 2023. Automated ICS template for STRIDE microsoft threat modeling tool. ACM Int. Conf. Proc. Ser. (January). https://doi.org/10.1145/3600160.3605068

Directive, C., 2008. 114/EC of 8 december 2008 on the identification and designation of european critical infrastructures and the assessment of the need to improve their protection. EUR-Lex.–2008.–23.12, L 345, 75–82.

DNP Users Group, 2005. A DNP3 Protocol Primer. Technical Report Revision A. DNP Users Group. Accessed: January 2026. https://www.dnp.org/Portals/0/AboutUs/DNP3%20Primer%20Rev%20A.pdf.

FIRST, 2023. Common vulnerability scoring system version 4.0 specification document https://www.first.org/cvss/.

Fujimoto, M., Itani, Y., Mitsunaga, T., 2023. Lessons learned for practical penetration test against industrial control systems. In: 2023 IEEE International Conference on Computing (ICOCO). IEEE, pp. 59–64.

Howard, M., Lipner, S., 2006. The security development lifecycle. https://doi.org/10.7765/9781526103482.00008

Khalil, S.M., Bahsi, H., Korotko, T., 2026. Penetration testing of newly-deployed industrial control systems. In: Proceedings of the International Conference on Critical Infrastructure Protection. Springer Nature Switzerland, pp. 25–48. First Online: 02 January 2026.

Khalil, S.M., Bahsi, H., Korõtko, T., 2024. Threat modeling of industrial control systems: a systematic literature review. Computers Secur. 136, 103543. https://doi.org/https://doi.org/10.1016/j.cose.2023.103543

Khalil, S.M., Bahsi, H., Ochieng'Dola, H., Korõtko, T., McLaughlin, K., Kotkas, V., 2023. Threat modeling of cyber-physical systems-a case study of a microgrid system. Computers & Secur. 124, 102950.

MITRE Corporation, 2023. Common attack pattern enumeration and classification (CAPEC). https://capec.mitre.org/. Accessed: January 2026.

MITRE Corporation, 2026. Common weakness enumeration (CWE). https://cwe.mitre.org/. Accessed: January 2026.

Modbus Organization, 2012. MODBUS Application Protocol Specification V1.1b3. Technical Report. Modbus Organization. Accessed: January 2026. https://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf.

OWASP, . Wstg - latest | owasp foundation. 2026 https://owasp.org/www-project-web-security-testing-guide/latest/.

OWASP, 2023. Owasp internet of things project. https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=IoT_Top_10. Accessed: 2023-04-30.

Penetration Testing Execution Standard, . Ptes technical guidelines. 2026 http://www.pentest-standard.org/.

Rapid7, 2026. Metasploit framework. https://www.metasploit.com/. Accessed: January 2026.

Saad, E., Mitchell, R., 2020. OWASP Web security testing guide version 4.2. The OWASP Foundation , 465.

Salzillo, G., Rak, M., Moretta, F., 2020. Threat modeling based penetration testing: the open energy monitor case study. ACM International Conference Proceeding Series . https://doi.org/10.1145/3433174.3433181

Searle, J., Rasche, G., Wright, A., Dinnage, S., 2016. NESCOR guide to penetration testing for electric utilities. National Electric Sector Cybersecurity Organization Resource (NESCOR).

Security for Industrial Automation and Control Systems - Part 2-3: Patch Management in the IACS Environment, 2015. IEC Technical Report IEC TR 62443-2-3. International Electrotechnical Commission (IEC).

Sharma, A., Sabharwal, S., Nagpal, S., 2023. A hybrid scoring system for prioritization of software vulnerabilities. Computers Secur. 129. https://doi.org/10.1016/j.cose.2023.103256

Stouffer, K., Pease, M., Tang, C., Zimmerman, T., Pillitteri, V., Lightman, S., Hahn, A., Saravia, S., Sherule, A., Thompson, M., 2023. Guide to Operational Technology (OT) Security. Technical Report NIST SP 800-82 Rev. 3. National Institute of Standards and Technology,Gaithersburg, MD, USA. https://doi.org/10.6028/NIST.SP.800-82r3

Strom, B.E., Applebaum, A., Miller, D.P., Nickels, K.C., Pennington, A.G., Thomas, C.B., 2018. Mitre att&ck: Design and Philosophy. In: Technical Report. The MITRE Corporation.

Süren, E., Fredrik, H., Olegård, J., Lagerström, R., 2023. PatrIoT: practical and agile threat research for IoT. Int. J. Inf. Secur. 22, 213–233. https://doi.org/10.1007/s10207-022-00633-3

Tharot, K., Duong, Q.B., Riel, A., Thiriet, J.M., 2023. Industrial cybersecurity game-scenarios based on the MITRE ATT&CK framework. Conference Proceedings - 2023 IEEE Asia Meeting on Environment and Electrical Engineering, EEE-AM 2023 , 1–4. https://doi.org/10.1109/EEE-AM58328.2023.10395155

ZigBee Alliance, 2015. ZigBee Specification. Technical Report Document 05-3474-21. ZigBee Alliance. Accessed: January 2026. https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf

Appendix 6

VI

T. Korõtko, S. M. Khalil, A. Schuller, V. Kotkas, J. Lambot, and K. Pappel. Implementation of MITRE CRUC for operational resilience assessment in cyber-physical systems. In *2026 IEEE International Conference on Industrial Technology (ICIT2026)*, 2026

Implementation of MITRE CRUC for Operational Resilience Assessment in Cyber-Physical Systems

1st Tarmo Korõtko
Dept. of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia
0000-0002-7368-1797

2nd Shaymaa Mamdouh Khalil
Dept. of Software Science
Tallinn University of Technology
Tallinn, Estonia
0000-0001-5487-1512

3rd Alvaró Schuller
Dept. of Software Science
Tallinn University of Technology
Tallinn, Estonia

4th Vahur Kotkas
Dept. of Software Science
Tallinn University of Technology
Tallinn, Estonia
0000-0003-2776-6033

5th Jaakob Lambot
Dept. of Electrical Power Engineering and Mechatronics
Tallinn University of Technology
Tallinn, Estonia

6th Karl Ivory Pappel
Dept. of Software Science
Tallinn University of Technology
Tallinn, Estonia

2026 IEEE International Conference on Industrial Technology (ICIT) | 979-8-3315-9875-4/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICIT64854.2026.11491056

***Abstract*—Cyber-physical systems (CPS) require empirically grounded methods to quantify resilience under cyber adversity. This study applies the MITRE Cyber-Resiliency Use-Case (CRUC) method on an instrumented CPS testbed by instantiating mission goals and weighted sub-objectives via the CREF Navigator, defining measurable actions, and computing a periodic resiliency index (R) and an integral Cyber Resiliency Quantification Index (CRQI). Two adversary scenarios were executed (denial-of-service against an Internet-facing web application and a pivot to the OT layer disrupting monitoring) in a benchmark and an alternative design (with rapid Layer-1 segmentation). Segmentation substantially improved resilience: CRQI decreased (improved) by 67% in Scenario #1 and 58% in Scenario #2, with higher minimum resiliency and reduced downtime. Outcomes were most influenced by sub-objective weighting, unambiguous action formulas, and scenario design that enabled repeatable telemetry and comparable assessment. Overall, CRUC coupled with automated monitoring provides mission-centric evidence for quantifying CPS resilience, while remaining sensitive to case-specific weights, metric coverage, and monitoring fidelity.**

***Index Terms*—cyber-resiliency, cyber-physical systems, cyber resiliency use case method, MITRE CRUC**

I. Introduction

Cyber-physical systems (CPS) underpin vital, mission-critical services, so engineering them to anticipate, withstand, recover, and adapt under cyber adversity should be as central as traditional safety and reliability practices [1]. Because cyber disruptions can propagate into the physical domain, purely theoretical or qualitative analyses may obscure cascading effects and interdependencies. To produce decision-quality evidence on mission impact and mitigation effectiveness, representative adversary behaviours must be exercised on real systems or high-fidelity testbeds [2]. Prior CPS taxonomies and threat-modelling work [3], along with case studies connecting threats, assets, and consequences [4], highlight diverse attack paths and reinforce the need for empirical validation of defences rather than purely theoretical assessment.

Empirical validation requires quantitative resilience metrics for evaluating trade-offs between investment needs and governance and design choices [5], [6]. Common families include performance-over-time resilience curves/areas [7], [8], risk-/topology-based indices [5], and mission-focused expert assessments mapped to standards [9]. We adopt the MITRE Cyber-Resiliency Use-Case (CRUC) method because it is mission-centric, aligned with the NIST cyber-resiliency engineering framework, and provides traceability from adversary behaviours to implementable resiliency techniques and measurable evidence [1], [10]. A remaining gap is practical, step-by-step guidance for operationalising CRUC with automated telemetry and repeatable experiments in CPS testbeds [2], [11].

This paper examines the practical implementation of the MITRE CRUC method in CPSs. It contributes to the existing body of literature by detailing the end-to-end operationalisation steps (artefact instantiation, metric definition, telemetry collection, and repeatable adversary/operator procedures) often omitted in prior work. The study addresses three questions: (1) how is the MITRE CRUC method applied in CPS; (2) which implementation activities and artefacts influence the resulting resilience assessment; and (3) whether CRUC is suitable for operational, automated (telemetry- and monitoring-driven) resilience assessment, rather than purely manual analytical evaluation [1], [5].

The remainder of the paper is organised as follows. Section II describes the operationalisation of CRUC, the case study, and the resilience quantification. Section III details the CPS testbed and the experiment procedure. Section IV reports results, discusses practical applicability, and outlines limitations. Finally, Section V concludes the paper.

This research was funded through the NATO DIANA program, under which Goldilock Secure Ltd. contracted TalTech to perform the testing (according to Tender nr 3-30/37-1 of 31.05.2024). This research was supported by the project „Increasing the knowledge intensity of Ida-Viru entrepreneurship“ co-funded by the European Union (2021-2027.6.01.23-0034).

Authorized licensed use limited to: Tallinn University of Technology. Downloaded on May 04,2026 at 16:49:17 UTC from IEEE Xplore. Restrictions apply.

II. Methodology

This study applies the MITRE CRUC method [12] to quantify resiliency. CRUC organises the analysis into five components (Fig. 1). The programmatic and system use concept defines the system type (CPS, enterprise IT, large-scale processing environment, or platform IT) and specifies primary, secondary, and supporting missions and their criticality. The architecture component describes the high-level design and identifies external interfaces and dependencies. The threat model characterises adversaries whose actions could degrade mission execution and may include relevant non-adversarial hazards. The alternative solutions component enumerates candidate approaches, from single technologies or architectural choices to bundles of technologies, design decisions, and operational or maintenance changes. Finally, scores, metrics, and measures of effectiveness (MoE) define the scoring framework for assessing the baseline system and comparing alternatives.

We operationalise CRUC through a CPS case study and testbed: we define mission goals/objectives and weighted sub-objectives in CREF Navigator [13]–[15], derive mission-relevant scenarios via contextual risk analysis and threat modelling, and implement measurable actions with 0-5 scoring and automated telemetry [12].

A. Practical applicability and expertise requirements

Deploying CRUC in an industrial CPS is feasible, with the main cost drivers being mission scoping and parameterisation (objectives, weights, thresholds). Because missions may be implicit and evolving, we recommend iterative instantiation: begin with minimal mission-critical functions and high-signal metrics, validate telemetry and scenario definitions, then refine objectives, weights, and metrics. Weights can be elicited via pairwise comparisons, Delphi-style expert consensus, or similar methods.

CRUC requires domain expertise across the workflow: (i) Mission and operations owners define unacceptable mission degradation and acceptable recovery; (ii) System architects and engineers identify interfaces, dependencies, and feasible segmentation and restoration procedures; (iii) Threat modeling security analysts map adversary behaviors to mission-impact pathways and select representative scenarios; and (iv) Measurement and data engineers implement telemetry, define thresholds and time horizons, and validate signal trustworthiness. While these needs are not unique to CRUC, the method makes them explicit by enforcing traceability between mission intent, adversary behaviours, and measurable evidence.

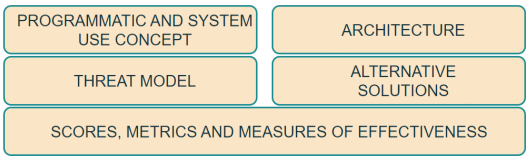


Fig. 1. The MITRE Cyber Resiliency Use Case Method [9].

We compute the cyber resiliency index R using the CRUC scoring method [9] (Equations 1 to 3), aggregating weighted actions (0-5) to sub-objectives and objectives.

$$PE^{SO} = \frac{100 \cdot \sum \left(PR^{A} \cdot PE^{A}\right)}{\sum \left(PR^{A} \cdot 5\right)} \tag{1}$$

In CRUC, each action performance element PE^{A} is scored on a discrete 0-5 scale, where 0 represents complete failure to meet the action intent, and 5 represents full satisfaction. Equation 1 therefore normalises the weighted sum of action scores by the maximum attainable weighted score, $\sum(PR^{A} \cdot 5)$, yielding PE^{SO} as a percentage.

$$PE^{O} = \frac{100 \cdot \sum \left(PR^{SO} \cdot PE^{SO}\right)}{\sum \left(PR^{SO} \cdot 100\right)} \tag{2}$$

$$R = \frac{100 \cdot \sum \left(PR^{O} \cdot PE^{O}\right)}{\sum \left(PR^{O} \cdot 100\right)} \tag{3}$$

To assess the system's overall cyber resiliency over time, the cyber resiliency index R is calculated periodically throughout the observed period. The calculated R values are used to calculate the Cyber Resiliency Quantification Index (CRQI) Γ using Equation 4, where n is the number of cyber resiliency values considered, i the index of cyber resiliency value 0, 1, …, $n-1$, R_i the cyber resiliency value at index i, and T_i the timestamp of cyber resiliency value.

$$\Gamma = \sum_{i=1}^{n} \frac{200 - R_{i-1} - R_i}{2} \cdot (T_i - T_{i-1}) \tag{4}$$

The Γ value is the trapezoidal approximation of the *resilience-loss area* (area between $R(t)$ and the ideal value of 100) over the observation window; smaller Γ indicates better resilience. When timestamps are in seconds, units are *score · seconds*. When a bounded index is needed, Γ can be normalised by the window length, but we report the unnormalised loss-area to preserve timing information and enable direct comparisons across scenarios of equal duration.

For the case study, several cyberattacks were designed and executed on the system under different scenarios. The resiliency values are recorded, and the CRQI value is calculated for all scenarios, and is used to conduct a comparative assessment between a benchmark scenario and alternative scenarios. By conducting a case study, it is possible to demonstrate how the MITRE CRUC method is applied in CPS, discuss which activities influence the calculation of the CRQI, and determine whether this method is suitable for operational resiliency assessment.

III. Experiment Design and Results

A. Hardware Setup

An experimental setup was established for the case-study, which emulates all relevant resources of the ICT system of the Manufacturing Company. The hardware and network diagram of the established ICT system are depicted on Fig. 2.

The system incorporates a layered, segmented network hierarchy with Enterprise (ENT), Operational (OT), and Field

Authorized licensed use limited to: Tallinn University of Technology. Downloaded on May 04,2026 at 16:49:17 UTC from IEEE Xplore. Restrictions apply.

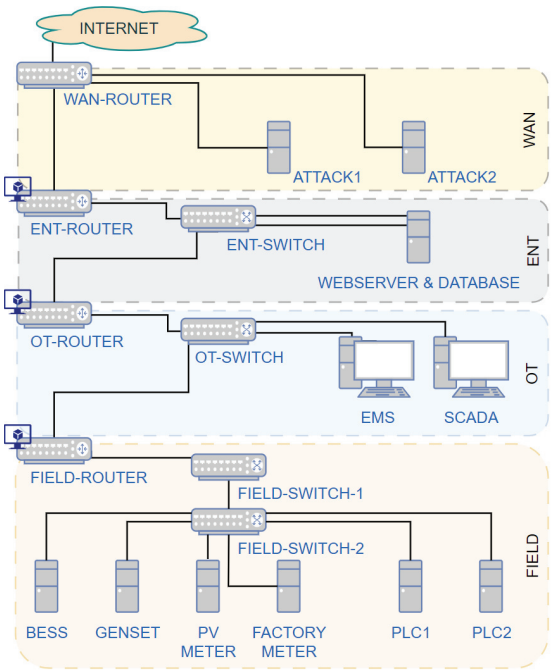


Fig. 2. ICT hardware diagram of the fictional manufacturing company.

(FIELD) layers, plus an isolated WAN layer for executing controlled cyberattacks. Two attacker hosts (ATTACK1 and ATTACK2) emulate public-origin traffic. Routing and firewalling are implemented as virtualised OPNsense instances with dedicated NICs, while a separate capture host records network traffic without interfering with operation. ENT hosts the Internet-facing application (web server and database). OT hosts supervisory and energy-management functions (SCADA and EMS). FIELD hosts representative controllers and meters (PLCs and power system instrumentation) that emulate the manufacturing process and supporting power system.

B. Resiliency Quantification

Using the CRUC method and the CREF Navigator tool [14], we selected the mission goal as *Withstand* and primary objective to *Continue*. Selected sub-objectives (weight 1-5) were: SO1 interruption (5), SO2 degradation (3), SO3 correctness (3); while supporting objectives were *Constrain*: SO4 isolate (5), SO5 change resource use (3); and *Recover*: SO6 restore functionality (5).

Parameter selection rationale: We assigned weights (1-5) to reflect relative mission impact in the case narrative per CREF/CRUC guidance. Thresholds and horizons were chosen to match the experiment time scale and keep actions comparable on the 0-5 scale. In practice, these weights must be carefully elicited with stakeholders and validated via bounded sensitivity checks. Threat modelling is used to select mission-relevant scenarios that anchor testing and metric design; the scenarios for this case study are described in Section III-C. We next define measurable actions for each selected sub-objective.

1) Interruption of Manufacturing (SO1): when the manufacturing control system produces expected outputs, the production of goods is not interrupted. The expected output is obtained by capturing data packages between the PLCs and the SCADA system. The interrupted manufacturing action $MI_j \in \{0, 1\}$ is calculated over a sliding window of i samples (ending at n) using Equation 5.

$$PE_{int_man} = \frac{\sum_{j=n}^{n-i} MI_j}{i} \cdot 5 \tag{5}$$

2) Interruption of Power Supply (SO2): when the grid is connected, and the fundamental grid parameters fall within grid code limits $\pm 10\%$, the resiliency component is at maximum. When the grid is unavailable, the power supply action is determined by considering the energy available from local resources (BESS, GENSET). The interruption of power supply is calculated using Equations 6 and 7, where GR is the grid availability status $\{0,\ 1\ \}$, F_{OK} denotes acceptable grid frequency $\{0,\ 1\}$, V_{OK} indicates grid voltage is within limits $\{0,\ 1\}$, B_{OK} is energy storage availability $\{0,\ 1\}$, E_B is the energy available in the battery [kWh], G_{OK} denotes if the generator is available $\{0,\ 1\}$, E_G is the energy available from the generator [kWh], E_{min} is the minimum energy required for maintaining the mission [kWh], and E_{max} denotes the energy required for maximum mission resiliency [kWh].

$$PE_{\text{int_pow}} = \left[GR \cdot F_{OK} \cdot V_{OK} + \frac{(1 - GR) \cdot E_{sys}}{E_{\max} - E_{\min}} \right] \cdot 5 \tag{6}$$

$$E_{sys} = B_{OK} \cdot E_B + G_{OK} \cdot E_G - E_{\min} \tag{7}$$

3) Degraded Communications (SO3): when the WEB_SERVER is unable to process requests from the WAN network, communications are considered as degraded. Equation 8 is used to calculate the action value proxied by WEB_SERVER availability $WSI_j \in \{0, 1\}$ over the last i responses (ending at n).

$$PE_{deg_comm} = \frac{\sum_{j=n}^{n-i} WSI_j}{i} \cdot 5 \tag{8}$$

4) Degraded Internet Connectivity (SO4): when a dedicated machine in the ENT layer is unable to connect to an external website. The action is calculated using Equation 9, where n is the index of the last request response, i is the number of requests considered and INT_j is the result of the Internet connectivity check $\{0,\ 1\}$ at index j.

$$PE_{deg_net} = \frac{\sum_{j=n}^{n-i} INT_j}{i} \cdot 5 \tag{9}$$

5) Manufacturing Integrity Compromised (SO5): packets that contain PLC firmware and software checksums are captured and the extracted values must match predefined values stored in a database. The action value is calculated using Equation 10, where n is the index of the last integrity

Authorized licensed use limited to: Tallinn University of Technology. Downloaded on May 04,2026 at 16:49:17 UTC from IEEE Xplore. Restrictions apply.

check, i the number of checks considered, m the number of manufacturing devices, and DI_j^k {0, 1} the result of integrity check for manufacturing device k at index j.

$$PE_{func_intg} = \frac{\sum_{j=n}^{n-i} \prod_{k=1}^{m} DI_j^k}{i} \cdot 5 \tag{10}$$

6) Power System Integrity Compromised (SO6): similar to the action "Manufacturing Integrity Compromised." Calculated using Equation 11, where n is the index of the last integrity check, i the number of checks considered, m the number of power system devices, and PSI_j^k {0, 1} the result of integrity check for power system device k at index j.

$$PE_{pow_sys_intg} = \frac{\sum_{j=n}^{n-i} \prod_{k=1}^{m} PSI_j^k}{i} \cdot 5 \tag{11}$$

7) Isolate Field Layer Devices (SO7): an operator emulates an intrusion detection system (IDS) by toggling a boolean flag, $ATCK$, to indicate whether the system is under attack. The toggling procedure must be explicitly specified in the test scenario. When $ATCK = false$ and $(T_{\text{det}} - t_{\text{cln}}) \leq T_{\text{now}} < (T_{\text{iso}} + t_{\text{cln}})$, the isolation action is computed using Eq. 12; otherwise, if $ATCK = true$ or $T_{\text{now}} < (T_{\text{det}} - t_{\text{cln}})$, it is computed using Eq. 13. Here, T_{det} is the latest $ATCK$ transition from $false$ to $true$, T_{iso} the latest transition from $true$ to $false$, T_{now} the current time, and t_{cln} and t_{reg} are user-defined cleanup and regression times. After detection, resiliency decreases. Once the attack is isolated and contained ($ATCK \rightarrow false$), the action value remains steady during cleanup and increases if no further attacks are observed, reflecting the completion of scan/restore/patch activities.

$$PE_{\text{iso_res_1}} = \frac{(T_{\text{iso}} - T_{\text{det}}) + (T_{\text{now}} - T_{\text{det}} - t_{\text{cln}})}{t_{\text{reg}}} \cdot 5 \tag{12}$$

$$PE_{\text{iso_res_2}} = \frac{T_{\text{now}} - T_{\text{det}}}{t_{\text{reg}}} \cdot 5 \tag{13}$$

8) Curtail Loads to Preserve Energy (SO8): Mission resiliency degrades as available energy decreases (e.g., depleted BESS capacity or GENSET fuel). The action is computed using Eq. 14, where E_B is battery energy storage capacity [kWh], E_G generator energy [kWh], P_{plant} plant power consumption [kW], and t_{pow_res} the required power reserve for maximum mission resiliency [s].

$$PE_{pow_avail} = \left(\frac{E_B + E_G}{P_{plant}} - t_{pow_res} \right) \cdot 5 \tag{14}$$

9) Functional Availability (SO9): The metric comprises two components: manufacturing availability and power-system availability. For each, packets are recorded containing process-status and power-status information. The action value is computed using Eqs. 15–18, where t_{rec_tol} is the interruption threshold that triggers a full manufacturing shutdown [s], t_{rec} the recovery duration [s], t_{man_halt} the duration of the most recent manufacturing stop [s], t_{pow_out} the duration of the most recent power outage [s], T_{now} the current time, and T_{man_stop} and T_{pow_lost} the timestamps of the latest $false \rightarrow true$ transitions for manufacturing stop and power loss, respectively.

$$PE_{\text{func_avail}} = \frac{t_{\text{rec_tol}} - t_{\text{rec}}}{t_{\text{rec_tol}}} \cdot 5 \tag{15}$$

$$t_{\text{rec}} = \max\left(t_{\text{man_halt}}, t_{\text{pow_out}}\right) \tag{16}$$

$$t_{\text{man_halt}} = \begin{cases} 0, & \text{if } MP_{\text{stop}} = 0, \\ T_{\text{now}} - T_{\text{man_stop}}, & \text{if } MP_{\text{stop}} = 1, \\ 0, & \text{if } PS_{\text{out}} = 0, \end{cases} \tag{17}$$

$$t_{\text{pow_out}} = \left\{ T_{\text{now}} - T_{\text{pow_lost}}, \text{if } PS_{\text{out}} = 1 \right. \tag{18}$$

C. Experiment Procedure

We follow an adversity timeline [16], [17] (Fig. 3) with modes normal (N), degraded (D), and faulty/lost (F). Two scenarios describe actions of the adversary (AD) or red team and the system operator/owner (IT - IT department, PM - plant manager, EE - electrical engineers, PO - process operator) or blue team for Scenarios #1 and #2 in Table I. Experiment Scenario #1 is a denial-of-service (DoS) attack on the web application exposed to the Internet, while Experiment Scenario #2 disrupts the monitoring capabilities of the OT client, where an external attacker pivots from the Internet to the OT layer.

Action durations are case-specific and should be engineered with stakeholders in practice. Here, durations follow the authors' interpretation for repeatability. For both scenarios, the alternative design replaces baseline procedures with rapid Layer-1 segmentation and restoration, used for isolating ENT/OT/FIELD layers and relevant assets (WEB_SERVER in #1; EMS/SCADA in #2).

IV. Experiment Results and Discussion

Experiments recorded benchmark and alternative runs for both scenarios. Cyber resiliency values and summary results are shown in Figs. 4 and 5 and in Table II. The trajectories show that the alternative design sustains a higher $R(t)$ during the adverse period and returns to nominal sooner, yielding a smaller loss area Γ in both scenarios. Overall, the case study shows how CRUC can be operationalised in a CPS: mission

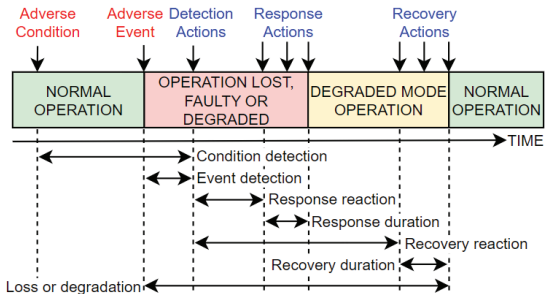


Fig. 3. Temporal decomposition of an adversity and system state transitions.

Authorized licensed use limited to: Tallinn University of Technology. Downloaded on May 04,2026 at 16:49:17 UTC from IEEE Xplore. Restrictions apply.

TABLE I
LIST OF ACTIONS FOR EXPERIMENT SCENARIOS

Nr	State	Action
		Experiment Scenario #1
1	N	AD builds network of bots
2	N	AD selects the WEB_SERVER as a target
3	N	AD prepares distributed bots for DDoS attack
4	F	AD initiates the attack
5	F	IT alerted about website downtime; traffic spike identified
6	F	IT analyses network traffic and registers an adverse event
7	F	IT initiates DDoS response and disconnects WAN network
8	F	IT separates OT and ENT networks as a precaution
9	F	IT isolates the WEB_SERVER machine
10	F	IT registers the adverse event as contained
11	D	IT contacts the ISP and requests traffic filtering
12	D	IT analyses network logs and deploys firewall and geoblock
13	D	IT restores connection to WAN network
14	D	IT validates firewall rules and tries 3 reconnect cycles
15	D	IT restores connection to WEB_SERVER machine
16	D	IT validates the WEB_SERVER and network traffic
17	D	IT restores connection between OT and ENT networks
18	N	IT registers system operation as NORMAL
		Experiment Scenario #2
1	N	AD gains access to the ENT layer
2	N	AD achieves code execution on the WEB_SERVER
3	N	AD obtains insecurely stored credentials
4	N	AD gains access to EMS machine and installs malware
5	F	AD executes malware on EMS machine
6	F	PM detects abnormal power consumption and contacts PO
7	F	PO stops manufacturing for safety concerns
8	F	PO communicates the issue to EE
9	F	EE reports everything OK with physical power system
10	F	PO realises the problem is in ICT systems and contacts IT
11	F	IT troubleshoots communications. Verifies everything OK
12	F	IT inspects EMS; detects malware; declares incident
13	F	IT disconnects WAN network. Problem persists
14	F	IT separates OT, ENT and FIELD networks
15	F	IT isolates the EMS machine
16	F	IT registers the adverse event as contained
17	D	IT collects evidence from the EMS machine
18	D	IT restores EMS from backup
19	D	IT restores EMS data; restart; reconnect
20	D	IT restores connection between FIELD and OT networks
21	D	IT, PO and PM verify the problem is resolved
22	D	PO restarts manufacturing
23	D	IT, PO and PM monitor system behaviour
24	D	IT restores connection between ENT and OT networks
25	D	IT, PO and PM monitor system behaviour
26	D	IT restores connection to the WAN network
27	N	IT registers system operation as NORMAL

goals and weighted sub-objectives are linked to measurable actions, automated telemetry provides the inputs, and $R(t)$ and CRQI (Γ) enable repeatable comparisons between design alternatives.

Assessment quality depended primarily on (i) sub-objective selection and weighting, (ii) measurable actions with unambiguous formulas, and (iii) engineered attack/response timelines and segmentation procedures. These decisions drove both instantaneous resiliency and the integral CRQI, shaping conclusions about mitigation efficacy. In this study, strict real-world fidelity of scenarios was less critical because the outputs are characterisation indices rather than directly interpretable engineering quantities. More advanced testing would incorporate human actors and physical intervention, replacing scripted scenarios with organisational procedures and practices.

1) Sensitivity to weights and thresholds: Because R is a hierarchical weighted aggregation of action scores, both $R(t)$ and the integral loss area Γ are sensitive to objective/sub-objective/action weights, as well as to thresholds and time horizons in the action formulas. For operational use, we recommend treating weights as bounded ranges (not fixed values) and running sensitivity checks by perturbing weights within plausible bounds (with renormalisation). In this case study, the mitigation (rapid Layer-1 (dis)connect) mainly affects availability- and isolation-related actions (e.g., SO1, SO2, SO4) that carry high weights, so the observed separation between benchmark and alternative scenarios should persist unless mission priorities change substantially. The same sliding-window and regression/cleanup parameters were used for both runs to preserve comparability (Section II).

2) Threat models where CRUC-based assessment may be less effective: CRUC-based operational assessment is least effective when objectives and actions fail to capture the dominant mission-impact mechanisms. For example, "low-and-slow" adversaries that stay within normal signal ranges or insider attacks changing logic without significantly affecting chosen telemetry. Also, scenarios where impact is driven by humans or organisational processes. Operational CRUC relies on trustworthy monitoring, and if telemetry can be tampered with, the resulting R and Γ may be incorrect.

3) Avoiding bias toward Layer-1 segmentation: To avoid unintentionally biasing the experiments toward Layer-1 segmentation, we kept the attack workload, system configuration, monitoring, and operator decision logic constant across benchmark and alternative runs. The only intended change was the isolation/restoration mechanism (rapid Layer-1 (dis)connect versus baseline procedures), which alters the duration of specific actions in the adversity timeline while leaving action definitions unchanged. This isolates the effect of faster segmentation on $R(t)$ and Γ, but also highlights a limitation: measured gains are tightly coupled to assumed operator procedures and timing parameters, which should be validated with stakeholders in real deployments.

TABLE II
EXPERIMENT RESULTS FOR EXPERIMENT SCENARIOS #1 AND #2

	Experiment Scenario #1		Experiment Scenario #2	
	Benchmark Scenario	Alternative Scenario	Benchmark Scenario	Alternative Scenario
CRQI	42 256	13 836	41 242	17 398
Lowest Cyber Resiliency Value	80.8	94.3	58.6	78.9
Cyber Resiliency Improvement	–	67%	–	58%
No Internet or manufacturing stopped	16m 00s	11m 30s	27m 15s	20m 55s

Authorized licensed use limited to: Tallinn University of Technology. Downloaded on May 04,2026 at 16:49:17 UTC from IEEE Xplore. Restrictions apply.

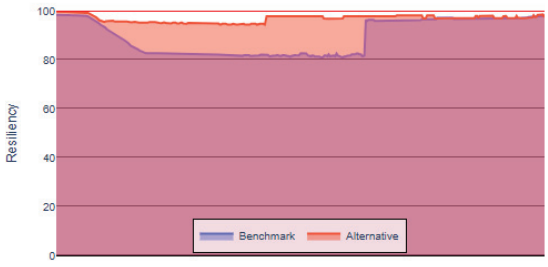


Fig. 4. $R(t)$ for benchmark, Scenario #1.

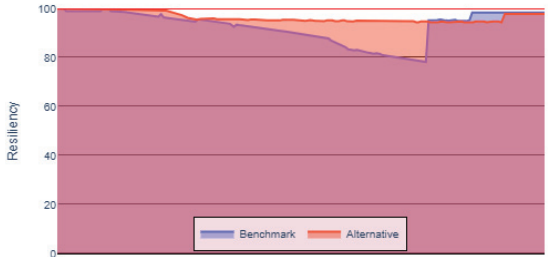


Fig. 5. $R(t)$ for benchmark, Scenario #2.

4) Relation to alternative assessment approaches: CRUC complements rather than replaces other resilience assessment approaches. Resilience-curve/area metrics concisely summarise performance loss and recovery, while risk- and topology-based indices support broader exposure analysis. CRUC adds mission traceability and a structured mapping from adversary behaviours to resiliency techniques and measurable evidence, supporting governance and engineering decisions. Our $R(t)$ and Γ computations can be viewed as resilience-curve metrics derived from CRUC-defined actions; a direct experimental comparison with other state-of-the-art operational methods is left for future work.

The results indicate CRUC is suitable for operational resiliency assessment when coupled with automated telemetry and repeatable scenarios. It detected and quantified benefits from Layer-1 segmentation: CRQI decreased by 67% (Scenario #1) and 58% (Scenario #2), with higher minimum resiliency and reduced downtime. However, results remain sensitive to case-specific weights, metric coverage, and monitoring fidelity, making careful metric and parameter design essential. Considering the aforementioned aspects, the CRUC method is deemed suitable for operational quantification of CPS resilience.

V. Conclusion

This paper operationalised the MITRE CRUC method on an instrumented CPS testbed, from mission scoping and threat-informed prioritisation to automated telemetry and computation of $R(t)$ and CRQI (Γ). Across two adversary scenarios, rapid Layer-1 segmentation improved operational resilience, reducing Γ by 67% and 58% and increasing minimum resiliency with reduced downtime. The most consequential implementation choices were the selection and weighting of sub-objectives, the specification of measurable actions with explicit formulas, and the design of attack/response timelines and segmentation procedures, which directly shape the instantaneous R values and the integral CRQI. The method is suitable for operational assessment when supported by sophisticated system analysis and robust telemetry, but outcomes remain contingent on context-specific weighting, completeness of metrics, and monitoring fidelity. Future work should broaden metric coverage, incorporate human-in-the-loop operations, and validate transferability across heterogeneous CPS domains to strengthen generalizability and governance utility.

References

[1] R. Ross, V. Pillitteri, R. Graubart, D. Bodeau, and R. McQuaid, "Developing cyber-resilient systems: A systems security engineering approach," National Institute of Standards and Technology, Special Publication 800-160, Volume 2, Revision 1, 2021.

[2] M. J. Weisman, A. Kott, J. E. Ellis, B. J. Murphy, T. W. Parker, S. Smith, and J. Vandekerckhove, "Quantitative measurement of cyber resilience: Modeling and experimentation," *arXiv preprint*, 2023.

[3] H. Bahsi, H. O. Dola, S. M. Khalil, and T. Korõtko, "A cyber attack taxonomy for microgrid systems," in *IEEE System of Systems Engineering Conference (SoSE)*, 2022.

[4] S. M. Khalil, H. Bahsi, H. O. Dola, T. Korõtko, K. McLaughlin, and V. Kotkas, "Threat modeling of cyber-physical systems—a case study of a microgrid system," *Computers & Security*, vol. 124, p. 102950, 2023.

[5] S. Hosseini, K. Barker, and J. E. Ramirez-Marquez, "A review of definitions and measures of system resilience," *Reliability Engineering & System Safety*, vol. 145, pp. 47–61, 2016.

[6] D. Bodeau, R. Graubart, R. McQuaid, and J. Woodill, "Cyber resiliency metrics, measures of effectiveness, and scoring (mitre technical report no. mtr180314). the mitre corporation," 2018.

[7] A. A. Ganin, E. Massaro, A. Gutfraind, N. Steen, J. M. Keisler, A. Kott, R. Mangoubi, and I. Linkov, "Operational resilience: concepts, design and analysis," *Scientific Reports*, 2015.

[8] G. Cybenko, "Quantifying and measuring cyber resiliency," in *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security, Defense, and Law Enforcement Applications XV*, vol. 9825. SPIE, 2016, pp. 156–161.

[9] R. D. Graubart, R. McQuaid, and J. Woodill, "Cyber resiliency metrics and scoring in practice," 2018.

[10] I. Mohamed, H. A. Hefny, and N. R. Darwish, "Enhancing cybersecurity defenses: A multicriteria decision-making approach to mitre att&ck mitigation strategy," *International Journal of Computer Networks & Communications*, vol. 16, no. 4, 2024.

[11] S. M. Khalil, H. Bahsi, and T. Korõtko, "Threat modeling of industrial control systems: A systematic literature review," *Computers & Security*, vol. 136, p. 103543, 2024.

[12] D. J. Bodeau, R. D. Graubart, R. M. McQuaid, and J. Woodill, "Cyber resiliency metrics catalog: Use case methodology and examples," The MITRE Corporation, Bedford, MA, Tech. Rep., 2018.

[13] MITRE, "ATT&CK for ICS matrix," Online resource. [Online]. Available: https://attack.mitre.org/matrices/ics/

[14] MITRE, "CREF navigator," Online resource. [Online]. Available: https://crefnavigator.mitre.org/navigator/

[15] MITRE, "ATT&CK enterprise matrix," Online resource. [Online]. Available: https://attack.mitre.org/matrices/enterprise/

[16] D. Firesmith, "System resilience: What exactly is it?" CMU/SEI Insights Blog, Nov. 2019. [Online]. Available: https://insights.sei.cmu.edu/blog/system-resilience-what-exactly-is-it/

[17] D. Firesmith, "System resilience part 3: Engineering system resilience requirements," CMU/SEI Insights Blog, Jan. 2020. [Online]. Available: https://insights.sei.cmu.edu/blog/system-resilience-part-3-engineering-system-resilience-requirements/

Authorized licensed use limited to: Tallinn University of Technology. Downloaded on May 04,2026 at 16:49:17 UTC from IEEE Xplore. Restrictions apply.

Curriculum Vitae

1. Personal data

Name	Shaymaa Mamdouh Mohammed Radwan Khalil
Date and place of birth	1981, Cairo, Egypt
Nationality	Egyptian

2. Contact information

Address	Tallinn University of Technology, School of Information Technologies, Department of Software Science, Ehitajate tee 5, 19086 Tallinn, Estonia
E-mail	shaymaa.khalil@taltech.ee

3. Education

2021–2026	Tallinn University of Technology, School of Information Technologies, Department of Software Science, PhD
2017–2020	Tallinn University of Technology and University of Tartu, School of Information Technologies, Cybersecurity (Digital Forensics), MSc, *cum laude*
2012–2013	University Paris-Dauphine and University Paris 1 Panthéon-Sorbonne, MBA International Paris
2006–2007	Synergy Professional Services, Egypt, professional training, scholarship of the Ministry of Communications and IT, graduation project: securing network traffic and access control on Cisco routers and switches
1998–2006	Cairo University, Faculty of Engineering, Department of Electronics and Communication Engineering, BSc, graduation project: Orthogonal Frequency Division Multiplexing (OFDM) simulation using MATLAB
1985–1998	Collège de la Mère de Dieu, Cairo, Egypt, General Secondary Education Certificate (Thanaweya Amma) and Alliance Française diploma

4. Language competence

Arabic	native
English	fluent
French	fluent
Estonian	beginner

5. Professional employment

2021–2026	Tallinn University of Technology, Estonia, Early Stage Researcher
2020–2021	Broadcom, Estonia, Technical Support Engineer
2016–2017	Orange Business Services, Egypt, Cloud System Administrator
2015–2016	Orange Business Services, Egypt, Security Coordinator
2015	Orange Business Services, Egypt, Change and Problem Manager
2012–2015	Orange Business Services, Egypt,

	Security TechPro and Technical Operation Owner (TechPro role now titled Security Manager)
2010–2012	Orange Business Services, Egypt, Security Operations Center, Security Support Engineer
2008–2010	Orange Business Services, Egypt, IT Service Operations, Customer Technical Support Specialist
2006	Cyber Egypt, Egypt, Customer Technical Support

6. Voluntary work

2012–2013	INJAZ, Egypt, Entrepreneurship Instructor
2000–2003	Resala, Egypt, Computer Courses Coordinator

7. Computer skills

- Operating systems: Windows, Linux
- Document preparation: LaTeX, MS Office, LibreOffice, Vim
- Programming languages: C++ (basic)
- Scientific packages: NVivo

8. Honours and awards

- 2020, Best Paper Award, "Analysis of Windows 10 Hibernation File", 6th Interdisciplinary Cyber Research Conference (ICR 2020), Tallinn University of Technology
- 2020, MSc in Cybersecurity, *cum laude*
- 2010, 2015, 2016, Local Award, Orange Business Services, Egypt
- 2009, Best Performer, IT Delivery, Orange Business Services, Egypt

9. Defended theses

- 2020, Analysis of Windows 10 Hibernation File, MSc, supervisors Dr. Hayretdin Bahşi and Pavel Tšikul, Tallinn University of Technology, School of Information Technologies, Department of Software Science
- 2013, Analyse de l'offre Cloud Computing d'Orange Business Services et du marché Cloud en Égypte, MBA, University Paris-Dauphine and University Paris 1 Panthéon-Sorbonne

10. Field of research

- Cybersecurity
- Industrial Control Systems Security
- Threat Modeling
- Digital Forensics

11. Supervised dissertations

- 2026, Grete Ohak, Enhancing Small Business Cybersecurity with Multi-Layer Threat Blocking and Retrospective Hunting, MSc, Tallinn University of Technology, School of Information Technologies, Department of Software Science
- 2026, Thomas Fankhauser, Evaluating the OWASP IoT Security Testing Guide: A Case Study on Consumer IoT Penetration Testing, MSc, Tallinn University of Technology, School of Information Technologies, Department of Software Science
- 2025, Andres Pelešev, Cybersecurity Regulatory Challenges in Small Modular Reactor Implementation: A Case Study of Estonia, MSc, Tallinn University of Technology, School of Information Technologies, Department of Software Science
- 2024, Leonard Benedikt Walter, Evaluating the Use of MASQUE Proxies for Achieving DNS Privacy, MSc, Tallinn University of Technology, School of Information Technologies, Department of Software Science
- 2024, Diego Del Rio Manzanas, Securing Remote Connectivity: A Systematic Penetration Testing Analysis of a Telepresence Robot, MSc, Tallinn University of Technology, School of Information Technologies, Department of Software Science
- 2023, Fothul Karim Forhan, Analysis of Infostealer Malware Samples and Proposed Defensive Measures, BSc, Tallinn University of Technology, School of Information Technologies, Department of Software Science
- 2023, Siim-Alexander Kütt, Stride-Based Black Box Penetration Testing Methodology for Micro-Mobility Services, MSc, Tallinn University of Technology, School of Information Technologies, Department of Software Science
- 2023, Helena Ingermann, Windows Subsystem for Android – Forensic Analysis, MSc, Tallinn University of Technology, School of Information Technologies, Department of Software Science
- 2022, Triin Viitmaa, Forensic Analysis of Windows Subsystem for Linux on Windows 11, MSc, Tallinn University of Technology, School of Information Technologies, Department of Software Science

12. Scientific work
Papers

1. S. M. Khalil, H. Bahsi, and T. Korõtko. Bridging industrial control systems design and testing through threat modeling-driven penetration testing – a microgrid case study. *Computers & Security*, 167:104906, 2026
2. T. Korõtko, S. M. Khalil, A. Schuller, V. Kotkas, J. Lambot, and K. Pappel. Implementation of MITRE CRUC for operational resilience assessment in cyber-physical systems. In *2026 IEEE International Conference on Industrial Technology (ICIT2026)*, 2026
3. S. M. Khalil, H. Bahsi, and T. Korõtko. Penetration testing of newly-deployed industrial control systems. In J. Staggs and S. Shenoi, editors, *Critical Infrastructure Protection XIX: 19th IFIP WG 11.10 International Conference, ICCIP 2025, Revised Selected Papers*, volume 763 of *IFIP Advances in Information and Communication Technology*, pages 25–48, Cham, 2026. Springer

4. S. Ali, A. Chub, H. Bahsi, and S. M. Khalil. Extending cyber threat modeling with physical dependency analysis in DC microgrids. In *2025 IEEE 66th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)*, pages 1–10, 2025

5. M. Erbas, S. M. Khalil, and L. Tsiopoulos. Systematic literature review of threat modeling and risk assessment in ship cybersecurity. *Ocean Engineering*, 306:118059, 2024

6. S. M. Khalil, H. Bahsi, and T. Korõtko. Threat modeling of industrial control systems: A systematic literature review. *Computers & Security*, 136:103543, 2024

7. S. M. Khalil, H. Bahsi, H. O. Dola, T. Korõtko, K. McLaughlin, and V. Kotkas. Threat modeling of cyber-physical systems - a case study of a microgrid system. *Computers & Security*, 124:102950, 2023

8. H. Bahsi, H. O. Dola, S. M. Khalil, and T. Korõtko. A cyber attack taxonomy for microgrid systems. In *2022 17th Annual System of Systems Engineering Conference (SOSE)*, pages 324–331, 2022

9. S. M. Khalil, H. Bahsi, and P. Tšikul. Analysis of Windows 10 hibernation file. In *6th Interdisciplinary Cyber Research Conference (ICR 2020)*, page 5, Tallinn, Estonia, 2020

Conference presentations

1. **S.M. Khalil**, H. Bahsi, and T. Korõtko. Penetration Testing of Newly-Deployed Industrial Control Systems. *Critical Infrastructure Protection XIX, 19th IFIP WG 11.10 International Conference*, Washington, D.C., USA, March 2025.

2. **S.M. Khalil**, H. Bahsi, and P. Tšikul. Analysis of Windows 10 Hibernation File. *6th Interdisciplinary Cyber Research Conference (ICR 2020)*, Tallinn University of Technology, p. 5, 2020.

Elulookirjeldus

1. Isikuandmed

Nimi	Shaymaa Mamdouh Mohammed Radwan Khalil
Sünniaeg ja -koht	1981, Kairo, Egiptus
Kodakondsus	Egiptus

2. Kontaktandmed

Aadress	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduse instituut, Ehitajate tee 5, 19086 Tallinn, Eesti
E-post	shaymaa.khalil@taltech.ee

3. Haridus

2021–2026	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduse instituut, doktoriõpe
2017–2020	Tallinna Tehnikaülikool ja Tartu Ülikool, Infotehnoloogia teaduskond, Küberturvalisus (digitaalforensika), MSc, *cum laude*
2012–2013	Pariisi Dauphine'i Ülikool ja Pariisi 1 Panthéon-Sorbonne'i Ülikool, MBA International Paris
2006–2007	Synergy Professional Services, Egiptus, erialane koolitus, side- ja infotehnoloogiaministeeriumi stipendium, lõpuprojekt: võrguliikluse turvamine ja pääsu reguleerimine Cisco ruuterite ja kommutaatorite abil
1998–2006	Kairo Ülikool, Inseneriteaduskond, Elektroonika ja sidetehnika osakond, BSc, lõpuprojekt: ortogonaalse sagedusjaotusega multipleksimise (OFDM) simuleerimine MATLABis
1985–1998	Collège de la Mère de Dieu, Kairo, Egiptus, üldkeskhariduse lõputunnistus (Thanaweya Amma) ja Alliance Française'i diplom

4. Keelteoskus

Araabia keel	emakeel
Inglise keel	kõrgtase
Prantsuse keel	kõrgtase
Eesti keel	algtase

5. Teenistuskäik

2021–2026	Tallinna Tehnikaülikool, Eesti, nooremteadur
2020–2021	Broadcom, Eesti, tehnilise toe insener
2016–2017	Orange Business Services, Egiptus, pilvesüsteemide administraator
2015–2016	Orange Business Services, Egiptus, turvakoordinaator
2015	Orange Business Services, Egiptus, muudatuste ja probleemide haldur
2012–2015	Orange Business Services, Egiptus,

	Security TechPro ja tehniliste toimingute omanik (TechPro ametikoha praegune nimetus on turvajuht)
2010–2012	Orange Business Services, Egiptus, turbekeskus, turvatoe insener
2008–2010	Orange Business Services, Egiptus, IT-teenuste käitlus, tehnilise klienditoe spetsialist
2006	Cyber Egypt, Egiptus, tehniline klienditugi

6. Vabatahtlik töö

2012–2013	INJAZ, Egiptus, ettevõtluse koolitaja
2000–2003	Resala, Egiptus, arvutikursuste koordinaator

7. Arvutioskused

- Operatsioonisüsteemid: Windows, Linux
- Dokumendivalmistamine: LaTeX, MS Office, LibreOffice, Vim
- Programmeerimiskeeled: C++ (algaja)
- Teadustarkvara paketid: NVivo

8. Autasud

- 2020, parima artikli auhind, "Analysis of Windows 10 Hibernation File", 6. interdistsiplinaarne küberuuringute konverents (ICR 2020), Tallinna Tehnikaülikool
- 2020, MSc küberturvalisuses, *cum laude*
- 2010, 2015, 2016, kohalik tunnustusauhind (Local Award), Orange Business Services, Egiptus
- 2009, parima töötaja tunnustus (Best Performer), IT Delivery, Orange Business Services, Egiptus

9. Kaitstud lõputööd

- 2020, Analysis of Windows 10 Hibernation File, MSc, juhendajad Dr. Hayretdin Bahşi ja Pavel Tšikul, Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduse instituut
- 2013, Analyse de l'offre Cloud Computing d'Orange Business Services et du marché Cloud en Égypte, MBA, Pariisi Dauphine'i Ülikool ja Pariisi 1 Panthéon-Sorbonne'i Ülikool

10. Teadustöö põhisuunad

- Küberturvalisus
- Tehnojuhtimissüsteemide turvalisus
- Ohtude modelleerimine
- Digitaalforensika

11. Juhendatud lõputööd

- 2026, Grete Ohak, Enhancing Small Business Cybersecurity with Multi-Layer Threat Blocking and Retrospective Hunting, MSc, Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduse instituut
- 2026, Thomas Fankhauser, Evaluating the OWASP IoT Security Testing Guide: A Case Study on Consumer IoT Penetration Testing, MSc, Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduse instituut
- 2025, Andres Pelešev, Cybersecurity Regulatory Challenges in Small Modular Reactor Implementation: A Case Study of Estonia, MSc, Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduse instituut
- 2024, Leonard Benedikt Walter, Evaluating the Use of MASQUE Proxies for Achieving DNS Privacy, MSc, Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduse instituut
- 2024, Diego Del Rio Manzanas, Securing Remote Connectivity: A Systematic Penetration Testing Analysis of a Telepresence Robot, MSc, Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduse instituut
- 2023, Fothul Karim Forhan, Analysis of Infostealer Malware Samples and Proposed Defensive Measures, BSc, Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduse instituut
- 2023, Siim-Alexander Kütt, Stride-Based Black Box Penetration Testing Methodology for Micro-Mobility Services, MSc, Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduse instituut
- 2023, Helena Ingermann, Windows Subsystem for Android – Forensic Analysis, MSc, Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduse instituut
- 2022, Triin Viitmaa, Forensic Analysis of Windows Subsystem for Linux on Windows 11, MSc, Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, Tarkvarateaduse instituut

12. Teadustegevus

Teadusartiklite, konverentsiteeside ja konverentsiettekannete loetelu on toodud ingliskeelse elulookirjelduse juures.

ISSN 2585-6901
ISBN 978-9916-80-606-7