Master Thesis

An approach to security testing in the context of smart power grids

Ole Bendik Midtbust

Thesis submitted for the degree of
Master in Informatics: Information Security
60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Spring 2020

Master Thesis

An approach to security testing in the context of smart power grids

Ole Bendik Midtbust

Master Thesis

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

The power infrastructure is undergoing a significant modernization, exposing the grid to new threats. The increasing competency of both public and private entities requires new approaches to the security testing of critical infrastructure. The increase in both size and complexity of the modern power grid allows for a more efficient and fault tolerant grids. However, this development introduces new challenges as sensors and legacy devices that were previously manually managed, are networked and controlled remotely. This technological shift within the domain of power infrastructure and the dependency of a stable power supply within a modern society presents new challenges within power grid security.

Organizations such as Enisa and NIST publish a significant amount of information relevant in the setting of software security, we propose an approach to security testing of a smart grid system utilizing information from bodies of knowledge to facilitate the testing. This thesis presents the approach and evaluates its feasibility within the domain of smart power infrastructure.

Our results show the feasibility of our approach in the context of the system the evaluation was applied to, but limitations in the setting of our trial hinders the assessment of its feasibility within the complete domain of smart grid.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisors Aida Omerovic from Sintef and Ketil Stølen from Sintef and the University of Oslo who have assisted me through this thesis. I am grateful for all the support Aida Omerovic has given, both in critical and constructive feedback as well as motivation and guidance to finish my thesis.

Chapter 1

Introduction

1.1 Motivation

Power grids are undergoing significant modernization. This results in efficient and fault tolerant grids. Larger grids represent compelling targets for malicious actors. With the higher competency of both public and private entities, securing such critical infrastructure from attacks is crucial. With the dependencies of a modern society on a working power grid, an attack that disables, either partly or completely, the power supply of an area could end up having devastating consequences.

An attack on the power grid, can in a worst-case scenario result in significant economic losses as well as the possibility for loss of life. The effect of such an attack was observed in Ukraine 2015, when three different power distribution companies had a power outage resulting in 225.000 people without power for a duration of several hours[1, 2]. Events such as this highlight the need for better and periodic security testing of smart-devices, and the importance of securing equipment connected to power grid infrastructure.

Security has traditionally been an afterthought in the development process of *ICT*-systems (Information Communication Technologies), but due to the criticality of a power grid, the distributed nature, and requirements of the power infrastructure, the development of secure systems must be prioritized. With the increasing capabilities and willingness from malicious actors, securing infrastructure that indirectly has the possibility to take down communication (both wired and cellular) in an area as well as disrupting emergency services and water supply is something that must be done properly. Whilst the smart grid can create a more efficient and robust power grid, it can also make other services more vulnerable due to most actors' dependency of a functional and reliable power supply[3].

As development continues in the field of smart grid technologies, and the use of smart devices in already established infrastructure sectors, the possibility for either accidents or faults in these complex systems demands

comprehensive security testing and validation. The power infrastructure is built up of a magnitude of different sensors and legacy devices. They were never intended or designed for being used in such an interconnected network as the smart grid represents[3]. This creates new challenges regarding securing both the physical and the software side of such a system, as well as the requirements for interfacing with older systems.

Smart grid is a new and continuously developing part of the modern power infrastructure, that is often built upon legacy-systems, and the need for security is increasing. With both the rise in complexity and number of dependencies of such a system, faults and vulnerabilities become more widespread and difficult to detect[3]. Something that does not seem important by itself, might in combination with vulnerabilities in other parts of a complex smart grid system be a way for a malicious actor to harm the grid in addition to harming both the resilience and reliability of the grid.

The objective of this thesis is to develop an approach to security testing within the domain of smart grid. The thesis is built upon a trial of the approach in the setting of emerging technologies within the realm of smart grid and critical infrastructure. The testing is done in order to create a standardized approach for testing of smart grid devices, as well as the system as a whole. Discovering possible vulnerabilities and their placement within a system is crucial for securing the smart grid and further development within the field.

The main results of the thesis include:

- An overview of the *«State of the art»*.
- A proposal of an approach to testing.
- An illustration of how the approach can be applied within the domain.
- A summary of the lessons learned when developing and trying out the approach.
- Recommendations for future work.

1.2 Contributions

This thesis presents three main contributions: 1. An overview of the state of the art. 2. An approach to security testing of a smart grid system. 3. A feasibility study of our approach in the setting of a proposed system.

1. An overview of the state of the art We have presented an overview of the state of the art relevant to the security testing of a smart

grid system (presented in Chapter 4). This contribution consists mainly of bodies of knowledge presenting, classifying, or otherwise categorizing the information. The information is in the form of recommendations, specific vulnerabilities, general weaknesses, checklists, and risk analysis approaches.

2. An approach to security testing of a smart grid system. Our approach (presented in Chapter 5) contains two main segments:

- A general process.
- A modelling approach.

The general process contains five distinct steps where the involved actors describe a system through the use of a set of graphical models presenting the relationship between the different components and possible vulnerabilities. Additionally, the process introduces an approach to use existing information and tools from several *bodies of knowledge* to assist with the testing and vulnerability mitigation of the components.

The modelling approach describes the requirements and specifications of four created models which are utilized by the involved actors to discover and present information about the system and its vulnerabilities. The models are sequentially created and used throughout the general process to assist the involved actors.

3. A feasibility study of our approach in the setting of a proposed system. Our third contribution in the form of a feasibility study is the application of the designed approach and its testing with the use of a trial of the approach (presented in Chapter 6) in the setting of a *Smart Grid* system with self-healing properties.

1.3 Thesis overview

The thesis consists of seven chapters described individually.

Chapter 1 - Introduction is divided into the following sections: Section 1.1 describing the motivation behind our thesis. Section 1.2 describing the contribution of our thesis. Section 1.3 describing the structure and giving a short description of all the chapters our thesis is comprised of.

Chapter 2 - Success Criteria presents the problem addressed by our thesis along with the different success criteria.

Chapter 3 - Research Method is divided into the following sections: Section 3.1 explaining the two main branches of research according to Stølen and Solheim [4]. Section 3.2 describes the difference between *Quantitative* and *Qualitative* research methods. Section 3.3 presents eight different strategies used for evaluating research methods and their relationship to: *Generality*, *Precision*, and *Realism*. Section 3.4 presents the evaluation strategy applied in our thesis.

Chapter 4 - State of the art is divided into the following main sections: Section 4.1 presenting different bodies of knowledge relevant to our thesis. Section 4.2 presenting different risk analysis and testing methods, tools, and modelling approaches. Section 4.3 presents four different types of *triangulation* in the setting of *qualitative* research.

Chapter 5 - Approach describes our initial approach to security testing and is divided into the following sections: Section 5.0.1 describes the overall approach. Section 5.1 through Section 5.5 presents the different steps of the approach along with responsibility of the actors involved in an examination. Section 5.6 presents the requirements and description of the test documentation and test plan templates to be used in an examination.

Chapter 6 - Trial of the approach is divided into the following sections: Section 6.1 presents our approach to security testing in the setting of a smart grid system with self-healing properties. Section 6.2 presents the lessons learned through the design, development, and application of our approach.

Chapter 7 - Discussion is divided into the following sections: Section 7.1 describes different threats to the validity and generality of our approach. Section 7.2 presents a discussion of our thesis with respect to our *Success Criteria* described in Section 2.

Chapter 8 - Conclusion presents the conclusions of our thesis and includes directions for future work.

Chapter 2

Success Criteria

The purpose of this thesis is to contribute to the field of security testing on critical infrastructure, with a Smart Grid system in focus. The following problem statement outlines the problem addressed by this thesis:

«Provide an approach for testing of a Smart Grid, which is applicable in the context of critical infrastructure security.»

In order to achieve this goal, six success criteria were devised that should be fulfilled. These criteria are designed with the listed actors in mind:

Security Analyst, knowledgeable about security testing of ICT systems.

Domain Expert, knowledgeable about either development of ICT systems or the workings and challenges of a modern power grid in order to assist the Security Analyst.

SC1: *The testing approach is customized with respect to the specific needs of the Smart Grid Domain.* The approach must be designed with the requirements of a Smart Grid system in focus. With the increased scope and complexity of a Smart Grid in comparison to a «typical» ICT system, the testing approach shall be designed with the relevant challenges in mind.

SC2: *The testing approach must be resource and cost efficient.* For the approach to be of use, both the amount of resources and time required needs to be justifiable with regards to the amount of information gained by its use. With time constraints on development of a new system, the approach needs to efficient comparable to other already in use approaches for it to be a reasonable choice.

SC3: *The testing approach must be generic enough for it to be of use on selected segments of a Smart Grid.* The approach needs to be broad enough for it to be relevant on several segments of a Smart Grid system. It must be designed in such a way that it is not too narrow for the scope of a Smart Grid as a whole.

SC4: *The testing approach must be viable during piloting of a Smart Grid*. With the scope of a Smart Grid, the approach must be designed in such a way that it is usable for the security analyst during the piloting phase of the systems lifecycle. Piloting phase is in this setting used for the phase between implementation of a system and having the system in production.

SC5: *The testing approach must result in unambiguous and detailed tests*. The approach must result in tests that are easy to implement and specific enough to not be misinterpreted. The information gained is required to be detailed and all results must be reproducible.

SC6: *Use of the testing approach must result in useful information*. The information gained by using this approach needs to be useful for the security analyst in such a way that what is gained, easily can be used to assess the system. The approach needs to be designed in such a way that it can be used together with existing databases and knowledge about common vulnerabilities and faults for ease of use.

Chapter 3

Research Method

3.1 Classical and Technology Research

Stølen and Solheim [4] divides research into two branches, classical research and technology research. With the first describing what is usually referred to as «the scientific method». These terms are also loosely connected to divide between «basic research» and «applied research». Classical research strives to describe and improve the knowledge about the world, similarly to «basic research», while «technology research» is more specific in the way that it is used to solve a specific problem, or improve a current solution which is more in line with the definition of «applied research». Stølen and Solheim describes the result of technology research as «artefacts».

As seen in Figure 3.1 «Technology Research» is an iterative research method, with the same steps as «Classical Research»(CR). This iterative process is comprised by the same main steps as in CR with the three steps:

- Problem Analysis
- Innovation
- Evaluation

The process starts with a problem, or «need». This «need» is often explained in the form of several success criteria which describe what must be fulfilled. The next step, «innovation» consists of the development of one or more artefacts with requirements stated in the success criteria. These artefacts can either be completely new, or an improvement of already existing artefacts. At this stage it is assumed that the artefacts fulfill the requirements of the first stage, a hypothesis. This hypothesis is then tested in the last step, the «evaluation». If the result of this evaluation is satisfying enough, then the process is finished, and the artefacts created fulfill the need defined in the problem analysis good enough. If the result of the evaluation is unsuccessful, or not satisfying enough for the hypothesis

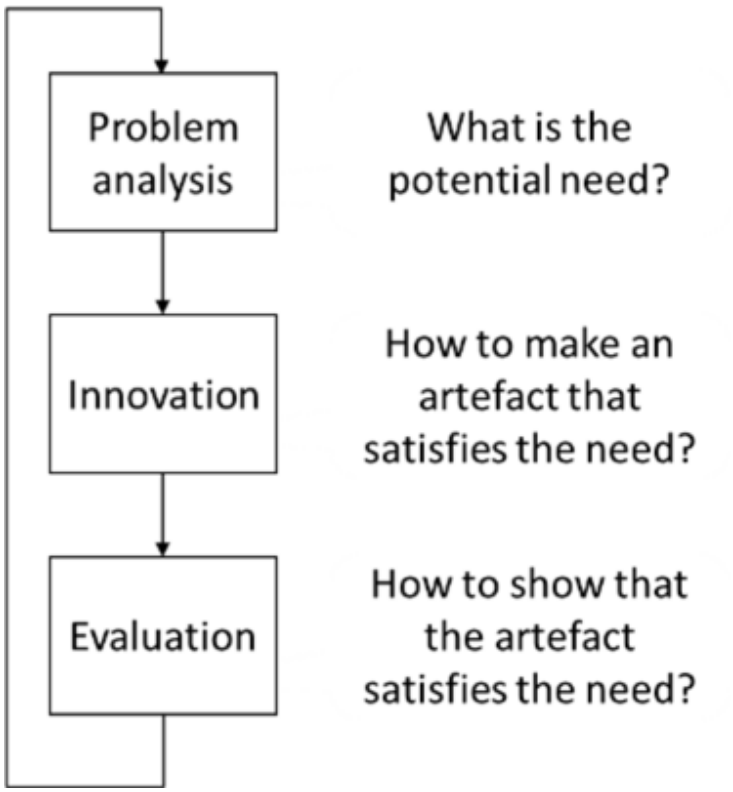

Figure 3.1: Main Steps in the method for technology research (adopted and modified from Stølen and Solheim, 2007 [4])

to either be completely, or partially fulfilled, the process starts again with a revisit to the first stage, the problem analysis. This iterative process can then be repeated until the results are deemed good enough with regards to the current problem analysis. [4, 5]

Nonetheless, it is important to note that any arbitrary development of artefacts is not necessarily «technology research», as without any new knowledge gained from the process, it is merely a case of technology development[4].

3.2 Qualitative and Quantitative Research Method

Research methods can according to Myers [6] be categorized as either «Qualitative» or «Quantitative».

- Qualitative research methods focus on observation and interviews in order to gain an understanding of the information gathered. According to Myers[6]: *«Qualitative research involves the use of qualitative data, such as interviews, documents, and participant observation data, to understand and explain social phenomena.»*

Examples of Qualitative research methods:

  - Action Research
  - Case Study Research
  - Ethnography
  - Grounder Theory

- Quantitative research focuses on number of data points and may be used in conjunction with mathematical models and laboratory experiments. This form of research was according to Myers[6] originally developed for use in natural sciences to study natural phenomenon.

  Examples of Quantitative research methods:

  - Survey Methods
  - Laboratory experiments
  - Formal methods (E.g. Econometrics)
  - Numerical methods (E.g. Mathematical models)

3.3 Evaluation Strategies

When evaluating a research method, McGrath[7] proposes to maximize:

- *Generalization*: Measures the validity of the method across several populations
- *Precision*: Measures the accuracy of the method
- *Realism*: Measures the similarity between the environment and reality

However, McGrath remarks that this creates a dilemma, as increasing either *Generalization*, *Precision*, or *Realism*, will always decrease either one or both of the other. This is described in Figure 3.2 where approaching e.g. Realism, takes you further away from both Precision and Generalization.

McGrath further describes eight research strategies (As shown in Figure 3.2):

- *Laboratory Experiment*, in an artificial setting increasing precision, but lacking in Realism and Generality.
- *Experimental Simulation*, a simulation of a specific setting from the real world, lacking in Generality and Realism.

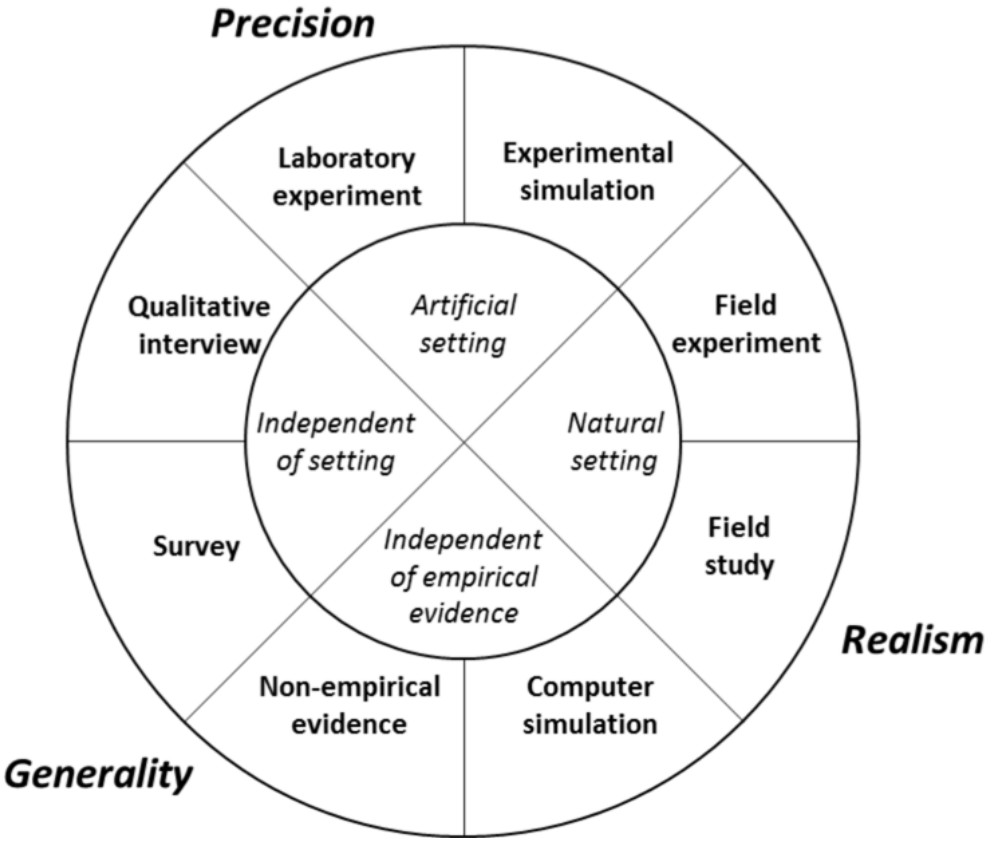

Figure 3.2: Evaluation strategies mapped with relation to *Precision*,*Realism*,and *Generality* (adopted and modified from McGrath, 1984 [7])

- *Field Study*, direct observation of a «natural» system, lacking in precision and generality, but strong on realism.
- *Field Experiment*, in a «natural» environment with input from researcher, stronger on precision than a Field Study.
- *Computer Simulation*, simulation of a specific setting gives increased Realism, but low Precision.
- *Non-Empirical Evidence*, based on logical reasoning, is without empirical evidence lacking in Realism and Precision.
- *Survey*, scores high on generality, but is difficult to control and bias from survey participants may reduce realism.
- *Qualitative Interview*, gives more precise answers than a Survey, but a lower number of participants reduces Generalization

The problem with these different properties (Generalization, Precision, and Realism) and lack of overlapping each other means according to Stølen

and Solheim[4] that it is required to choose several strategies. These strategies should be chosen is such a way that they complement each other for a greater total coverage. They further describe three additional important choices a researcher must decide on:

- *«Is the strategy feasible?»*
  Each strategy has a different «cost» associated, either in the form of availability of individuals, or the different monetary costs of an experiment vs. a computer simulation.

- *«How to ensure that a measurement really measures the property it is supposed to measure?»*
  The object or situation measured must be isolated, and the researcher should be able to account for as many factors as possible that can influence the result.

- *«What is needed to falsify the prediction?»*
  The strategy chosen must in the given situation allow for a negative outcome. A strategy that is guaranteed a positive result in the specific setting is described as worthless by Stølen and Solheim[4].

Stølen and Solheim[4] describes Evaluation Strategies as tools which give a researcher possibilities, but also limitations and constraints. A qualitative interview is not able to properly test a systems function. They claim that due to the possibility of falsification of the prediction to be used, the choice of strategy has already, to a certain degree, been decided.

3.4 Research method in the setting of our thesis

In this section we have described our use of technology research in the setting of our thesis based on the three steps presented in Figure 3.1: *Problem Analysis*, *Innovation*, and *Evaluation*.

Problem Analysis In Section 1.1 we presented our motivation behind this thesis and outlined a «need» to improve the security testing of the power grid and critical infrastructure. To better understand the domain, we collected information about the *State of the Art*, and presented an overview in Chapter 4. Based on the «need» and the state of the art, we devised a problem statement which we presented in Chapter 2: *«Provide an approach for testing of a Smart Grid, which is applicable in the context of critical infrastructure security.»* And based on the problem statement, established six *Success Criteria* our artefact should fulfill:

SC1: *The testing approach is customized with respect to the specific needs of the Smart Grid Domain.*

SC2: *The testing approach must be resource and cost efficient.*

SC3: *The testing approach must be generic enough for it to be of use on selected segments of a Smart Grid.*

SC4: *The testing approach must be viable during piloting of a Smart Grid.*

SC5: *The testing approach must result in unambiguous and detailed tests.*

SC6: *Use of the testing approach must result in useful information.*

Innovation Based on the *Success Criteria* described in Chapter 2 we designed an *Artefact*, in the form of an approach to the security testing of a smart grid system utilizing the different *bodies of knowledge* described in the State of The Art (Chapter 4). This approach was designed around the actors described in Chapter 2: a *Security Analyst* and *Domain Expert* and we presented this approach in Chapter 5.

Evaluation The third step of technology research requires the choice of one or several evaluation strategies to assess the fulfillment of the *Success Criteria* we presented in Chapter 2.

We required an evaluation strategy able to properly assess the fulfillment of our criteria based on their relationship to the different attributes we described in Section 3.3 (*Generality*, *Precision*, *Realism*). SC2 and SC5 describes specific attributes the result is required to have, and SC4 is in this case sufficiently covered by SC3. The choice of an evaluation strategy was thus based on the remaining success criteria.

SC1 requires an evaluation strategy with a high degree of *Realism* as the testing approach must fit the *«specific needs of the Smart Grid Domain»*.

SC3 requires an evaluation strategy with a high degree of *generality* as the testing approach must be general enough for it to be applicable on *«selected segments of a Smart Grid»*.

SC6 requires an evaluation strategy with a degree of both *Realism* and *Precision* as the results of the approach must *«result in useful information»* and must thus be applicable to the real world.

With respect to the relevant success criteria we could benefit from an evaluation strategy based around *Realism* and *Generality*. Figure 3.2 presents the relationship between *Generality*, *Precision*, and *Realism*

and the evaluation strategies presented in Section 3.3. SC1 and SC6 could benefit from an evaluation strategy in a natural setting, whilst SC3 could benefit from an evaluation strategy independent of either empirical evidence or setting (As described in Figure 3.2).

In accordance with the main objective of this thesis (As described in Chapter 2) the evaluation strategy chosen must in some way be applicable for all or most of the success criteria. As both SC1 and SC3 require some degree of *Realism*, either a *Field Experiment* or a *Field Study* could be beneficial to the evaluation of our artefact. These strategies both require a natural setting, but a study is stricter with regards to the interference from the researcher on the environment.

In Section 3.2 we presented two different research methods (quantitative and qualitative). In the setting of an evaluation strategy based around *Realism*, a qualitative method could be beneficial, as a quantitative is more fitting in a laboratory setting.

In the design of our evaluation, there were several factors to consider. The scope of our evaluation is limited both in time and personnel, and we are thus required to make some sacrifices with regards to our evaluation. Ideally, we would have evaluated our artefact in the setting of either a *Field Study* or *Field Experiment*, but this requires both significant time, preparations, and in-depth access to a fitting system. A compromise to this would be performing the evaluation on a system in an artificial setting, based on a general description of a smart grid system. This approach severely harms the *Realism* of our evaluation but allows us to test the feasibility of our artefact in a controlled setting, with a significantly lower resource requirement. Our evaluation strategy thus falls somewhere between a *Field Experiment* and *Experimental Simulation*. And we attempted to simulate (in an artificial setting) the feasibility of our artefact in the form of a trial of the approach (Chapter 6).

In Section 3.3 we presented Stølen and Solheim's[4] description of evaluation strategies, and how each strategy introduces limitations and constraints for the researcher. The setting of our strategy and its application on a described system as opposed to a real-world system set certain limitations on our evaluation. Our trial is based on the description of a system, and we are thus unable to: 1. Design tests applicable to the system. 2. Apply the tests in the setting of the system.

Additionally, our trial was limited through the scope of our thesis. Introducing constraints based on time and personnel, preventing the application of our approach in the setting of multiple real-world systems in addition to preventing the application of the complete approach. These limitations prevented us from assessing both the testing and mitigation of vulnerabilities, as well as properly illustrating the application of the different templates described in Section 5.6 designed to assist the involved actors in this process.

In summary, we applied our artefact in the setting of a system described

in [8] with the purpose of assessing the feasibility of our approach. We conducted a feasibility study in the setting of a trial from the perspective of a *Security Analyst*, in the context of an examination with the aim of discovering possible vulnerabilities within a smart grid system with self-healing properties. We presented this trial of the approach in Chapter 6 and the results of our evaluation regarding our *Success Criteria* in Chapter 7.2.

Chapter 4

State of the art

4.1 Bodies of Knowledge

There are several good tools describing the best practice both while developing, as well as maintaining and testing IoT systems exist. Some of the most well established are published by OWASP[9], Enisa[10], SANS[11], CIS[12] and NIST[13], who all regularly give out best practice recommendation in addition to working towards raising awareness and spreading information about security in the cyber realm. There are also foundations like MITRE, who focus more on the indexing and enumeration of both vulnerabilities and weaknesses through projects like *«Common Vulnerability and Exposure»* (CVE)[14], *«Common Weakness Enumeration»* (CWE)[15] and *«Common Attack Pattern Enumeration and Classification»* (CAPEC)[16].

4.1.1 MITRE's CVE, CWE, CAPEC, CWRAF

MITRE[17], an American non-profit organization started the project CVE[14] in an effort to categorize and index all discovered vulnerabilities in a standardized fashion. This effort later expanded into several other projects, among them, CWE[15] and CAPEC[16]. CWE, perhaps the most useful with regards to a smart grid, consists of information about the overlaying weaknesses that make vulnerabilities possible. This is useful as an easy metric to reference and compare the coverage of both tools and recommendations, which is the reason for MITRE's CWE Compatibility project. This project has as purpose to describe coverage of tools or services by using the standardized CWE list.

CVE, in contrast with CWE is less abstract and describes vulnerabilities in place of general weaknesses. These vulnerabilities, instead of being broad and including, are application specific and therefore not useful in a developer setting. The intent of the CVE entries is instead to create a collection of the different know vulnerabilities affecting systems, with

information about the weakness and possible mitigations. This makes CVE useful when maintaining and updating a system, or when performing security testing.

Another useful tool by MITRE with regards to security testing is their CAPEC project. This project is an attempt to describe a weakness from the view of a threat agent, and can be useful when testing, as some entries include detailed information about how the exploit works. The entries contain information about metrics like severity, likelihood and consequence, as well as references to either more abstract CAPECS entries, or the overlaying CWE that enables it. Some of these metrics, like consequence and likelihood, are not «fits all», and might thus not be applicable to every organization. These metrics should be decided in accordance with the affected organization through a risk assessment or a similar process.

In addition to these projects from MITRE, an effort has been made to rank the CWE in a similar way to projects like «OWASP Top Ten»[9], this is through a project called the «Common Weakness Risk Analysis Framework»(CWRAF)[18]. This CWRAF is a part of the CWE project, and makes it possible to define domains, or what MITRE refers to as «Vignettes», and rank weaknesses after importance in these different domains[19]. By using this framework, the CWE project and SANS[11] worked together compile a list of what they believed to be the «Top 25 Most Dangerous Software Errors»[20], it is however, important to note that this list has not been updated since 2011.

4.1.2 OWASP

OWASP[21] is an organization dealing with cybersecurity recommendations, specifically their «OWASP Top Ten» project[9]. This project is presented in the form of a list with what is regarded as the 10 most critical Web Application risks and includes detailed explanations of both the risk itself as well as possible ways to prevent or mitigate the weakness that makes it possible. Each entry on the list includes:

1. Examples of attacks, to easier explain how the vulnerability is exploited

2. Internal or external references, to both other OWASP resources and projects like MITRE's CWE to better explain the overlaying weakness or NIST guidelines to prevent it altogether

OWASP makes an effort not to recommend specific software to prevent the impression of being biased towards certain vendors. OWASP does also release what they call «OWASP Top Ten IoT»[22], it comes in the form of a poster, and is too lacking in specifics to be of use. It is more a list of general

recommendations than a comprehensive document like their well-known Top Ten project.

4.1.3 SANS

SANS is in contrast with OWASP a company, which in addition to giving best practice recommendations also offers payed services in the form of security training and certification[11]. They do however have several free resources in the form of research papers and weekly news. Among these is Securing Web Application Technologies (SWAT)[23], which identifies what SANS believes to be the minimum standard required to prevent vulnerabilities in your application. This SWAT comes in the form of a short checklist with seven categories with around 10 entries on each. This checklist is primarily for Web Applications, but some of the entries general enough that they are valid for other types of systems. In addition to the best practice recommendations and SWAT list they help maintain and annually publish what they refer to as the *«Top 25 Software Errors»*[20] where they list and references the *«25 most dangerous software errors»* in the form of CWE entries.

4.1.4 CIS

The Center for Internet Security[12], better known as CIS have developed what they call CIS Controls[24] or CIS Top 20. This is a list of actions developed to better protect both the organization and the data it is both processing and storing. It is designed in a way that makes it easily automated and ordered from 1-20 after importance. These 20 entries are sorted into several of what they call domains: «basic», «foundational», and «organizational». These domains are an effort to better separate the entries with regards to their «position» in a system. Each of these 20 entries can further be split down into «sub-controls», which contains details the specific sub-controls as well as about how they can be measured. Similarly to OWASP, CIS instead of recommending specific software or solutions, only present what they consider to be «best practice» recommendations.

4.1.5 Enisa

Enisa[10], an agency subject to the EU, has a primary focus on both the knowledge and expertise of cybersecurity concentrating on member states and companies operating in the EU. They assist both with policy making, as well as releasing publications and tools like their *«Enisa Good Practices for IoT and Smart Infrastructures Tool»*[25] which is both easy to use as well as highly relevant to Smart Grid Security and what

they refer to as *Industry 4.0* systems, or systems involved in the 4th industrial revolution. This tool is meant to be used for Risk assessment with information about what Enisa considers to be best practice with regards to e.g. «Authorization» and external references deemed to be relevant. The tool contains two subsections directly relevant to Smart Grid security which is utilized during the examination:

- Baseline security IOT
  - «Baseline cybersecurity recommendations with a focus on Critical Information technology equipment.»[3]
- Industry 4.0
  - «The main objectives were to collect good practices to ensure security of IoT in the context of Industry 4.0/Smart Manufacturing, while mapping the relevant security and privacy challenges, threats, risks and attack scenarios.»[26]

The tool contains information about the different domains a given topic is located in, as well as the possible threat groups. The tool is searchable, and Enisa recommends the tool to be used in conjunction with «Use-Case scenarios».

4.1.6 NIST

NIST[13] is an agency subject to the United States Department of Commerce. They primarily supply recommendations, tools, and publications to be used for expanding US industry. Additionally, the contain a significant number of resources relevant to the smart grid domain[27]. They have developed «the NIST Cybersecurity framework»[28], a framework designed to be used for securing critical infrastructure. The framework can however be used outside this domain with some customization, which is recommended as the framework is only designed for guidance. In addition to the mentioned framework and resources, they supply security standards publicly for the US government which can be used by others.

4.1.7 Relevance to a Smart Grid System

Of the projects and organizations mentioned in this section, there are primarily two types, the databases and lists supplied by MITRE and OWASP which are often referenced by others, and the best practice recommendations and tools supplied by SANS, CIS and Enisa. The databases like CVE, CWE and CAPEC are objective as they explain a specific weakness or vulnerability, but are meant to be referenced and thus difficult to use without the context given from other resources similar to

«OWASP Top Ten». When looking at recommendations from e.g. OWASP or SANS, it is important to recognize who is making the recommendations, as SANS is a for-profit organization and might thus be biased with regards to recommendations of either software or other solutions. Enisa and NIST however are both publicly funded, and their recommendations are either of their own free-to-use tools, or simply «best-practice».

With regards to Smart Grid security, there is a lot of information supplied by NIST and Enisa, both in the forms of tools from Enisa, and publications and frameworks from NIST. In addition to these agencies, MITRE also has «Power Domain» vignette[18] in their CWRAF project through the CWE database that gives information about the different components of a smart grid, as well as external references for further information.

4.2 Risk Analysis & Testing

The knowledge of where to find information about both vulnerabilities, weaknesses and recommendations both while testing, and developing a system is not enough. It is important for the user to manage these recommendations and the published information in an efficient way. This can be done with the use of Risk management and different Risk analysis methods and tools.

4.2.1 CRAMM & OCTAVE

With regards to cyber risk analysis methods we describe the «CCTA Risk Analysis and Management Method» (CRAMM) and «Operationally Critical Threat, Asset, And Vulnerability Evaluation» (OCTAVE).

CRAMM[29] is a methodology where the analysis focuses on risk identification and assessment while the management part focuses on identifying possible mitigations for these risks. The approach is divided into three steps, focusing on asset identification, threat and vulnerability identification as well as mitigations or countermeasure identification. The CRAMM method helps provides ISO17799 compliance and additionally fulfilling the documentation requirements for ISO27001.

OCTAVE[29] is built up similarly to CRAMM with three steps focusing on identification of assets and threat profiles, components and vulnerabilities, and lastly risks and risk mitigations. The OCTAVE approach in comparison to CRAMM, uses workshops and brainstorming for information gathering instead of the interview approach used by CRAMM.

4.2.2 Modelling

Modelling a system while focusing on the security and risk aspects requires both a different mindset as well as different approaches. There are several types of approaches that can be used for models in the security domain, and they can usually be divided into three categories[29]:

- Tree-Based
  - Attack-trees
- Table-Based
  - HAZOP
- Graph-Based
  - CORAS

Tree-based graphical models show causation and the ability to decompose an unwanted incident, be it in the form of an attack as seen in «Attack-trees»[30], or faults and malfunctions as described by «Fault-Trees» used in a Fault Tree Analysis (FTA)[31].

«Attack-trees» are comprised of several nodes, with a root node consisting of a malicious unwanted incident e.g. «Virus infect computer», and a hierarchy consisting of decomposed child nodes with unwanted incidents. The unwanted incident described by a node, is dependent on the incidents described in the connected child nodes to occur for it to be possible. «Attack-tree» supports conditions like 'OR' and 'AND'. 'OR' is the default state, and 'AND' can be used if several child-nodes are required to allow the parent-nodes incident. Each node in such a tree can contain information about likelihood, cost of action, and specific requirements such as special tools or knowledge. This makes an «Attack-tree» suitable for modelling specific attacks, as it «sees» the system from the attacker's viewpoint and contains a detailed map of how an unwanted incident is achieved.

A «Fault-Tree» is similar to the «Attack-trees» described above, with instead of the focus being on a malicious unwanted incident, it describes a unwanted incidents in the form of faults or errors in a system, concentrating less on the security aspect in favor of reliability and failure analysis. A «Fault-Tree», can similarly to the «Attack-tree» be implemented with basic logic gates (e.g. 'AND'/'OR').

4.2.3 HAZOP

HazOp[32] is a risk analysis method originally intended for industrial processes but is fitting in the domain of cyber security. It works by

sequentially examining either a working, or a planned process or flow, searching for possible deviations from the intended design similarly to the process used for misuse case diagrams. This is done with the use of «guide words» (e.g. 'More', 'Less', 'No/None') to describe what is necessary to fit the intended flow described in the design specifications. These guidewords are used in a table along with the relevant component and parameters.

4.2.4 CORAS

The CORAS[33] Method contains an easily understood language based on UML which is designed to facilitate understandable graphs for people with different backgrounds. The modelling language focuses on the harm a threat agent can do to an «Asset» through threat scenarios and unwanted incidents. It is constructed in a way to estimate both the likelihood and consequence of the risks described in the diagram as well as the vulnerabilities that should be mitigated. A CORAS diagram is constructed through the use of several different constructs, we present a simplified CORAS diagram in Figure 4.1.

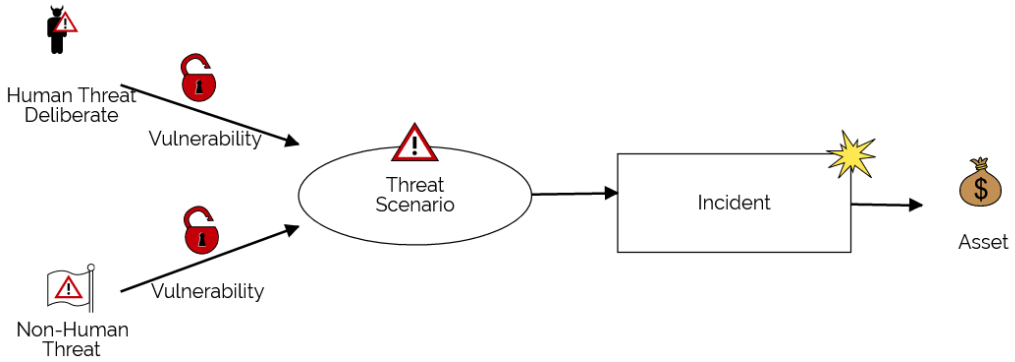

Figure 4.1: A *threat* exploiting a *vulnerability* creates a *threat scenario* allowing an *incident* harming an *asset* presented using the CORAS[33] language.

4.2.5 Testing Principles

According to Black, Veenendaal, and Graham [34], there exists seven fundamental testing principles in action on most projects.

- Testing shows the presence of defects
- Exhaustive testing is impossible
- Early testing
- Defect Clustering

- Pesticide paradox
- Testing is context dependent
- Absence-of-errors fallacy

These principles are depicted in Table 4.1 and describe in general terms the limits and pitfalls of software testing.

4.2.6 Testing Levels

Software testing is described by Black, Veenendaal, and Graham [34] as dividable into four distinct levels:

1. *Component Testing:* Testing of components, modules or units that are separately testable. This type of testing can be performed isolated from the rest of the system. This level can according to Black, Veenendaal, and Graham include testing of characteristics such as decision coverage, robustness, performance, resource behavior, and the functionality of a given component.

2. *Integration Testing:* This level focuses on the communication and interaction between the different components and parts of a system. This type of testing can according to Black, Veenendaal, and Graham be performed on a collection of components, database elements relevant to the components, system infrastructure, component interfaces, system configuration and other configuration data.

3. *System Testing:* Testing of the system or end-product as a whole. This may according to Black, Veenendaal, and Graham include; *«test based on risk analysis reports, system, functional, or software requirements specification, business processes, use cases, or other high level descriptions of system behavior, interactions with the operating system, and system resources»*[34]. Black, Veenendaal, and Graham describes this type of testing as the last step during development to make sure that the finished system matches up with the original specifications. It should include the testing of both functional and non-functional requirements.

4. *Acceptance Testing:* The Acceptance testing is performed after the development is complete. This level is according the Black, Veenendaal, and Graham usually the last level, and intended to answer the questions *«Can the system be released?»*, *«What, if any, are the outstanding (business) risks»* and *«Has development met their obligations?»*[34]. They further explain that the tests should be based on the user and system requirements, use cases, business processes and risk analysis reports. This is not necessarily the last

step and might be followed by another Integration test if a system is intended to be integrated with other systems on a larger scale.

4.2.7 Security Testing

There are several methods for security testing of a system, we will focus on Model-based security testing and Risk-Driven security testing. Model-based security testing can according to Schieferdecker, Grossmann and Schneider[35] be divided into four sub-groups: «Security functional testing, model-based fuzzing, risk- and threat-oriented testing and security test patterns». With each sub-group covering a different part of the testing.

- «Security functional testing» covers security implementations.
- «Model-based fuzzing» covers input validation.
- «Risk- and threat-oriented testing» is based on first identifying possible risks/threats and create test based on these.
- «Security test patterns» covers known attack-patterns which can be found through projects like CAPEC[16].

They further explain that risk driven security testing focuses more on the possible risks/threats, and design test cases based on these, as well as the consequence of the unwanted incident. This does however require a risk analysis in order to function as intended. In comparison to a model-based security testing, risk driven security testing focuses on what has the biggest consequence for a given system and design the tests to prevent this.

4.3 Triangulation

Triangulation is in our thesis important regarding the understanding of the system being tested, the creation of tests (both based on the creator, and the component being tested), mitigation of discovered vulnerabilities and the responsibilities of the involved actors.

The logic behind triangulation within qualitative studies is according to Patton[36] based on the thought that *«no single method ever adequately solves the problem of rival explanations»*. He further explains that as different methods reveal different aspects of reality and are affected by the inherent weakness of the specific method used, a combination of different methods, while more resource intensive, is better able to prove the consistency of the data, and the validity of the result.

Patton describes four types of triangulation:

- *Method Triangulation*

- *Investigator Triangulation*
- *Theory Triangulation*
- *Data Source Triangulation*

4.3.1 Method Triangulation

Patton[36] describes *Method Triangulation* as the use of several methods to answer different questions to better describe the same reality. He explains that this can be done by using both qualitative and quantitative methods in the form of a comparative analysis.[36]

4.3.2 Investigator Triangulation

Patton[36] describes *Investigator Triangulation* as the use of several investigators or observers in an effort reduce the inherent bias that comes with the use of a singular investigator. Using several investigators to independently investigate either a situation or the data, and later comparing their findings might prevent some interpretive bias and selective perception. People, based on their background and experience are likely to interpret the same data in different ways, possibly leading to different theories and conclusions.

4.3.3 Theory Triangulation

Patton[36] describes *Theory Triangulation* as investigating the same data with the use of several different theoretical perspectives. With the base idea being the understanding of how different assumptions and conditions affect the interpretation of data. An example of *Theory Triangulation* is using the viewpoints of different stakeholders when examining data, it is according to Patton common for different stakeholders to disagree about the goals and purpose of a program, and how this goal should be reached.

4.3.4 Data Source Triangulation

Patton describes *Data Source Triangulation* as *«comparing and cross-checking the consistency of information derived at different times and by different means within qualitative methods»*[36]. Patton goes on to describe that the intent of this type of triangulation is to get an understanding of when and why the data is different. A difference in the data gathered from different sources is not necessary an indication that one is right, and one is wrong. The difference is explained with the fact that different data sources can describe different aspects of the same reality.

Principle 1:	**Testing shows the presence of defects**	Testing can show that defects are present but cannot prove that there are no defects.
Principle 2:	**Exhaustive testing is impossible**	Testing everything is not feasible except for trivial cases.
Principle 3:	**Early testing**	To find defects early, testing activities shall be started as early as possible in the software or system development life cycle.
Principle 4:	**Defect Clustering**	Testing effort shall be focused proportionally to the expected and later observed defect density of modules.
Principle 5:	**Pesticide paradox**	If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects.
Principle 6:	**Testing is context dependent**	Testing is done differently in different contexts.
Principle 7:	**Absence-of-errors fallacy**	Finding and fixing defects does not help if the system built is unusable and does not fulfil the users' needs and expectations.

Table 4.1: The seven fundamental principles of testing, adopted and modified from Black, Veenendaal, and Graham, 2012[34].

Chapter 5

Approach

In this chapter we have presented an initial version of the proposed approach. This approach addresses what the different actors are expected to cover during the process of security testing a system. Certain steps of this approach present a set of requirements, rather than specific instructions for the actors to follow.

The approach leverages several directed graphs, where each node is a category relevant to a smart grid implementation. We use the structure of a smart grid distribution network as an example, focusing on the communication flows between the different components, as well as the *«Distributed Systems Operator»* (DSO) and the *«customer-edge equipment»* (CEE).

Each leaf-node in this graph contains references to either a document or a database entry from standards organizations, e.g. NIST[13] or MITRE's CWE[15]. The modelling approach is designed in such a way that several nodes can all point to the same leaf-node, and a single node can have connections to several sub-nodes. The graph is semi-hierarchical (directed non-cyclical structure, but a single node can have several parent-nodes) and contains three node-types; root-node, intermediate (simply referred to as 'node'), leaf-node. It does not differentiate the nodes outside of these three groups.

The approach is designed for use by security analysts on either an existing system, or on a system during development, with the intention to prevent unwanted incidents, and not to be used in a reactionary manner. The approach is focused on vulnerabilities relevant to Smart Grid and the problems that arise when combining *Operational Technologies* (OT), referring to the technologies used in industrial operations for management and monitoring of physical equipment and processes, and *Information and Communications Technology* (ICT), referring to the combination of traditional IT and communication technologies, -based technologies within the domain of smart grid and power distribution.

This approach is not focused on either intentional or accidental unwanted incidents but is instead approaching the issue from the vulnerabil-

ity side, as opposed to incident-oriented approaches. The idea is that this approach is something an analyst can use to get an indication of the different weaknesses that a given system contains, based on what components it is comprised of, the distribution of these weaknesses, and information about how these weaknesses can be mitigated.

5.0.1 Steps of Approach

The approach is comprised of two main segments, the general process and a modelling approach.

The process consists of five main steps as seen in Figure 5.1, where in each step, the actors, *Security Analyst* and *Domain Experts* defined in the *Success Criteria* in Chapter 2 have different roles and tasks. The steps have either different tasks for each actor, or a single task they are expected to cooperate on, to better utilize their different backgrounds and knowledge.

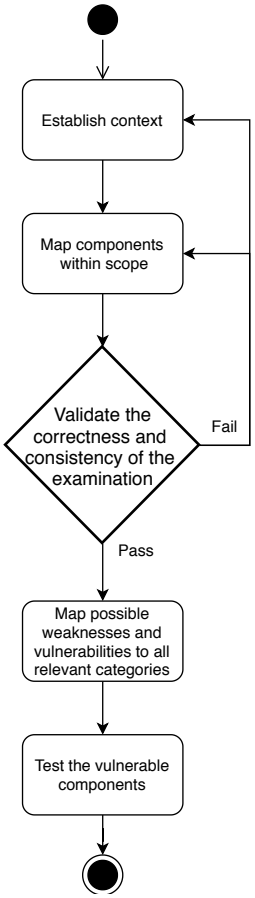

Figure 5.1: Diagram describing the flow between the different steps of the approach

The modelling approach consists of detailed instructions and rules for the *Security Analyst* to design and utilize a set of graphs and models to

assist the overall approach. The models created are expected to be used in a sequential manner, where the information gained from one, is used as a base for building the following models. The relationship between the different graphs and models has been described in Figure 5.2.

The Security Analyst The constant emergence of new threats[37] makes it necessary for the *Security Analyst* to stay updated on current trends and recently discovered vulnerabilities reported by organizations such as MITRE with their CVE system [14]. It is not feasible to have knowledge about vulnerabilities in every domain reported to either MITRE or other organizations, but it is required for the *Security Analyst* to have some knowledge about any new developments with regards to security of the specific components and technologies used as a way to gather information for the development of tests on specific test objects. The less technology and component specific bodies of knowledge, e.g. the resources published by the *The OWASP project*[21] or other bodies with reported information comparable to guidelines and *best practice* type resources, is able to assist the *Security Analyst*. Additional general knowledge within the domain of cyber security can aid the *Security Analyst* in avoiding common pitfalls or other security limitations relevant to the distinct domain.

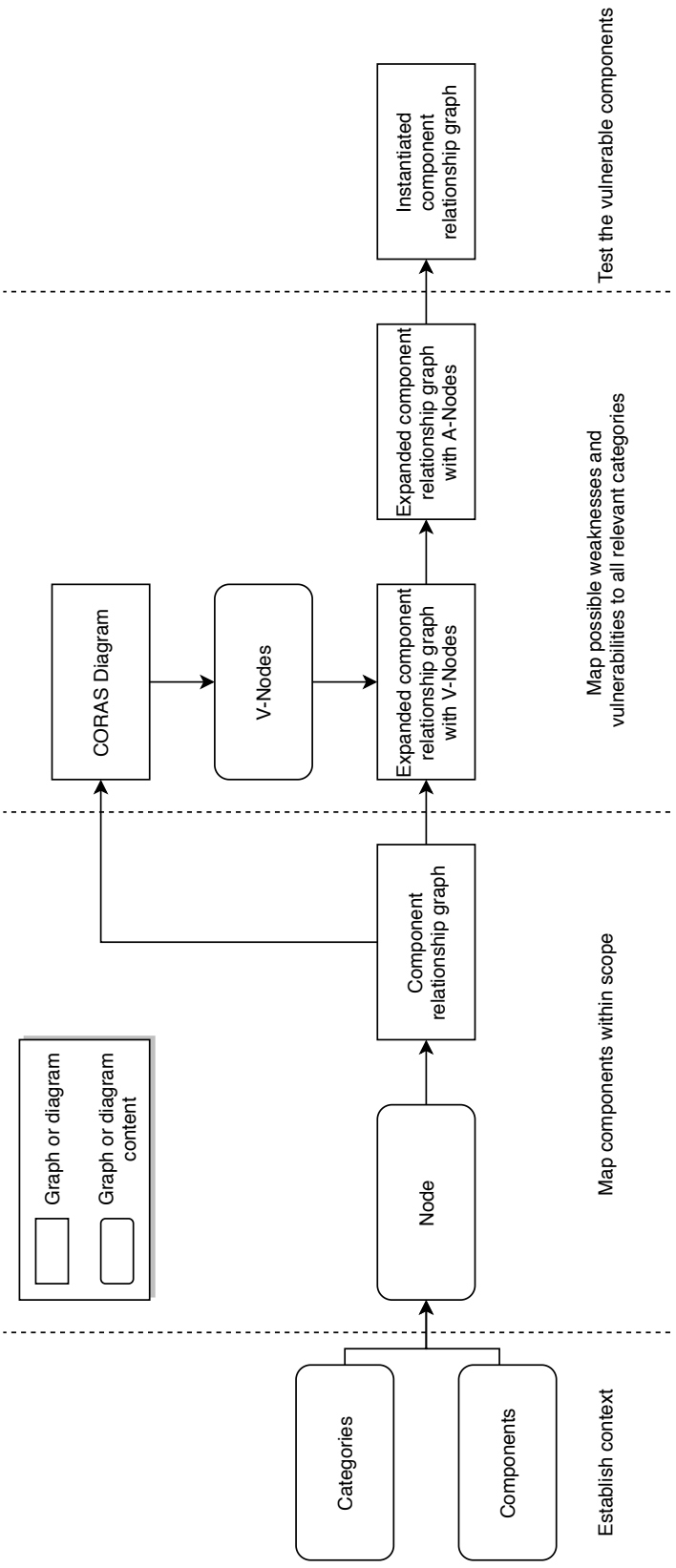

Figure 5.2: A conceptual model describing the relationship between the different graphs and their content. The has been divided into four segments corresponding to the steps of the approach where the models are created.

5.1 Establish context

For the approach to be of use, clear boundaries and expectations need to be made with regards to:

The scope of the examination

- What is the end-goal of the security testing of the system? The «Testing Goals» must be clearly defined for the *Security Analyst* to better focus the resources during the testing phase. Is the end-goal an improvement of the *Confidentiality*, *Integrity*, or *Availability* of the system?
- How much time and personnel are assumed to be used for the examination? A clear timeframe must be defined by the *Security Analyst* in order to efficiently plan the time usage on the different tests and to help with defining the boundaries of the examination.
- It is required to have information about the people involved in the examination, and which roles these people have. Are they involved with; The testing of components? The design of the required test-cases? The development of models and the grouping of the different component relationships? This responsibility of each person included in the examination must be clearly outlined by the *Security Analyst*.
- Which components are part of the examination and what defines the boundary which separates «within scope», and «outside of scope»? The *Security Analyst* must in collaboration with the *Domain Experts* list all components deemed «within scope», and part of the examination. All components described during the examination must be listed here, and all components listed here must be described later in the examination. This includes components that might, through a cascading fault, impact the *«main»*-components of the examination.

The assumptions done by the *Security Analyst* or the *Domain Experts*

- Any assumptions made with regards to the function and structure of the system to be examined must be listed in such a way that an actor not previously familiar with the system is able to partake in the examination. Any items listed here must be described in such a way that they are understandable for people with some domain knowledge and not necessarily involved with the examination.

The background and understanding of the domain the *Security Analyst* has

- There must be information listed about both what kind of domain knowledge the *Security Analyst* possesses, and both the professional and educational background of the *Security Analyst*. As the background of the *Security Analyst* has an impact of the examination, it is important to have a description of what the users of this approach has with regards to domain knowledge and experience within the field of critical infrastructure, smart grid, and other industrial systems, as well as within the general domain of cyber security.

The background and understanding of the domain the *Domain Experts* has

- There must be information listed about both what kind of domain knowledge the *Domain Experts* has, and both the professional and educational background of the different *Domain Experts*. Is the background from *Industrial Automation*-systems, a *Smart Grid*, or a different branch within *ICS* and *Critical Infrastructure*?

Examination is in this context used to describe the complete process, containing all steps described in the approach.

All components within the scope of examination must be categorized with regards to their use within ICT and OT relevant to a Smart Grid system by the *Security Analyst* with assistance from the *Domain Experts* with regards to the relationship between the different components. This categorization should be done with a focus on grouping components together based on common attributes and abilities. A component or category can belong to multiple categories, and a category can contain multiple components or sub-categories. The categorization is in this step presented in a list of all the categories, and which components or sub-categories it contains.

Ex:

1. <*Category*>: Contains <*Sub-category*> and <*Sub-category*>
2. <*Category*>: Contains <*Component*> and <*Component*>

If it is unclear how a set of components can be categorized, the *Security Analyst* may need to make assumptions based on; the functions of the components in the context of the complete system, the abilities of the component in the context of the complete system, the location of the component compared to other components, either physical or its placement within the system in the context of software and data communication. Any assumptions made here must be mentioned in the list described in the beginning of this section.

5.2 Map Components within scope

All components described in the previous step must be mapped based on their relationship to each other in a non-cyclical directed graph by the *Security Analyst*, there should be no need to involve the *Domain Experts* in this step, and rather have them validate the choices and structure in the next step. The mapping is built on the concept of what sub-part of the system being tested a chosen component, or abstraction belongs to. As described in step 1, a component or sub-category can be part of several categories, and a category can contain several sub-categories and components. The graph should be created in such a way that attributes relevant to a category should be valid for all sub-categories and components linked to this category. The categories, sub-categories and components are further on described as nodes. This step should result in a non-instanced general directed graph describing the system, and the relations between the components described in step 1.

The building of this graph consists of mapping the relationship between the different *Components*, *Categories*, and *Sub-categories* into a graph-structure with connections directed from the *Category* to the *Sub-category/Component*. Every unique *Category*, *Sub-category*, and *Component* should result in a single node. Performing this action on every entry of the relationship-list created in Section 5.1 results in either a single or several graphs, if the latter, create a *«Super-node»* with connections directed towards all the top-level nodes of the previously created graphs. This *«Super-node»* is named the same as the system being tested. Since the approach does not require an equal number of nodes on each sub-tree from the root-node, the *Security Analyst* should make an effort to order the different levels. We define all nodes with the same vertical height as parts of the same *level*. This ordering is done in such a way that all leaf-nodes are at the same level (the bottom), with a similar abstraction-level going upwards. If a *Category* describes a major part of the system, it should not be on the same level as a *Category* grouping two simple components together. As shown in Figure 5.3, it is not necessary for a level to contain nodes from each sub-tree, making it possible to *«skip»* levels if a node is not in the correct level of abstraction, or is otherwise a bad fit.

5.3 Validate the correctness and consistency of the examination

The created graph must be validated and checked for accuracy in the context of the real-world system by the *Domain Experts*, are all the choices made by the *Security Analyst* in the previous step consistent with the real-world system, and are the placements logical with regards

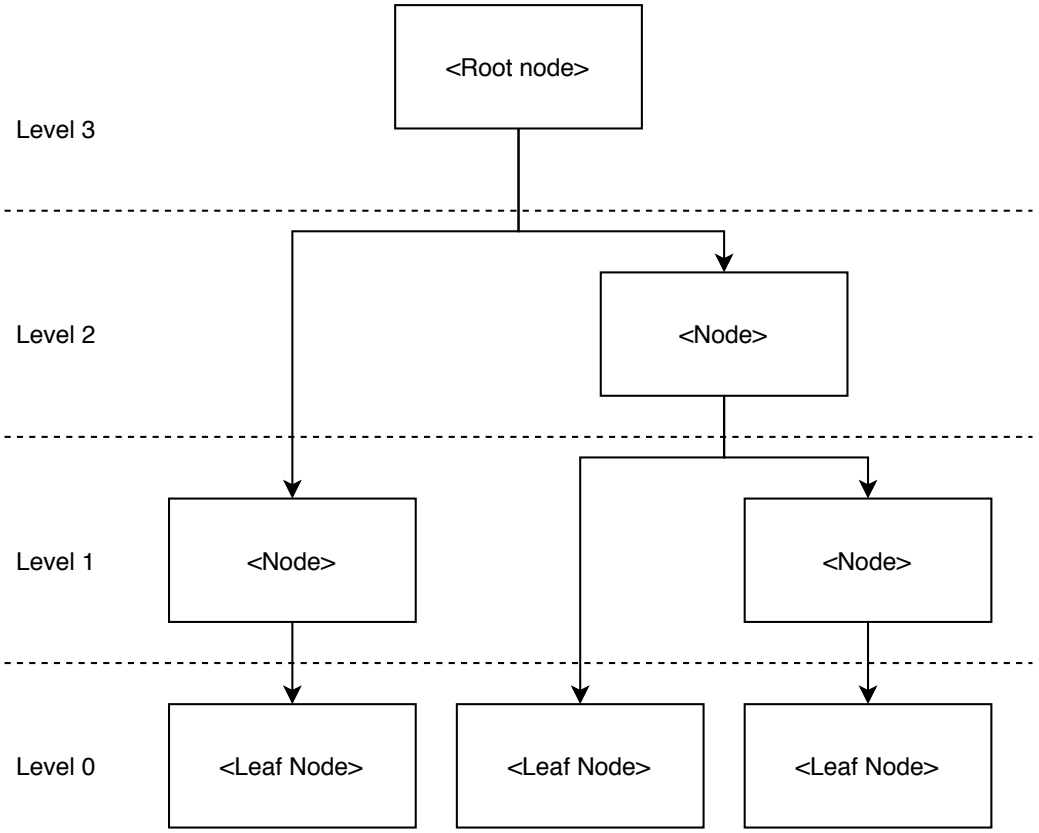

Figure 5.3: Figure depicting an example of the graph created in Section 5.2

to the existing system? The *Security Analyst* must validate the graph for correctness with regards to the syntactical structure of the graph as well as the relationship between the different nodes and leaf-nodes. All nodes should fulfill the requirement that all attributes belonging to a node, should be applicable to all connected sub-nodes. The graph must be non-cyclical as a category cannot indirectly inherit attributes from itself, and the connections created in the graph are one-way connections, all directed away from the root-node (top), in the direction of the leaf-nodes (bottom). The graph is required to be consistent with regards to choices made during the representation of components, as well as the connections between all the nodes. If a component in step 1 was grouped under specific categories or sub-categories, this needs to be reflected in the designed graph. Any irregularities and inconsistencies with these requirements must be amended by repeating either one or both of the previous steps according to Figure 5.1.

The *Security Analyst* is required to create strict rules defining what is, and what is not relevant to the examination. There needs to be rules substantiating the choices made with regards to the node groupings. These rules need to be provable when comparing the graph and the assumptions and mappings described in step 1. This is done in order to gain traceability

on the choices made and making it possible to later justify the design choices. Any inconsistencies and ambiguities discovered during this step must be adjusted by going back to the relevant step and revising either the design, assumptions, or the choices made as depicted in Figure 5.1.

The rules defined should be in the form of either:

1. <*Component or Category*> is within scope of the examination because <*reason*>.
2. <*Component or Category*> is outside of scope of the examination because <*reason*>.
3. <*Component or Sub-category*> is a part of <*Category or Sub-category*> because <*reason*>.

An effort should be made to keep these rules as simple as possible. The rules should for traceability contain a reference to either the document or specification the listed reasoning is based on. The *Domain Experts* are expected to assist in outlining the reason behind why a component is deemed inside or outside the scope of the examination.

Example in the context of a *Neighborhood Area Network (NAN)* equipped with *Smart Meters*:

1. The *Smart Meter* is within scope of the examination because the information communicated to the *Substation* has a direct effect on the localized power distribution as described in <*diagram*> and it acts as an intermediary between the *NAN* and *Home Area Network (HAN)* as described in <*document*>.
2. The Customer Equipment is outside of scope of the examination because it is not controlled by the *DSO* and the interface between the *HAN* and *NAN* is handled by the *Smart Meter* as described in <*diagram*>.
3. The *Neighborhood Gateway* is within scope of the examination because it acts as an intermediary between the *NAN* and the *Wide Area Network (WAN)* as described in <*document*>

5.4 Map possible weaknesses and vulnerabilities to all relevant categories

The *Security Analyst* is through the use of a misuse case[38] or CORAS diagram[33] expected to describe possible weaknesses and how they can be relevant to a specific system. The diagram is created with regards to the system and the relevant assets, where the *Security Analyst* maps the possible weaknesses and vulnerabilities to the relevant categories.

The described weaknesses are added to the type of model described in section 5.2 (Figure 5.3). This mapping results in the type of model presented in Figure 5.4. The mapped weaknesses, depicted as circular nodes on the graph connected with a dotted line, referred to as a Vulnerability-Node (V-Node), must follow the requirement of inheritance described in section 5.2. A weakness mapped to a category, is relevant for all sub-categories of that given category. These weaknesses or vulnerabilities, and the content of the *V-Nodes* is described in the form of statements. The content of the *V-node* is the vulnerability allowing the threat scenario to expand into an unwanted incident.

Ex: «Use of weak or outdated crypto-algorithms»

These statements are required to be short, easy to read, and understandable in such a way that little or no additional information is required to understand the general form of the weakness.

The mapping is executed with a bottom-up approach, where the *Security Analyst* checks if the V-Node is relevant to a component, and if that is the case, checks the V-Nodes validity with regards to the connected node one abstraction-level higher, referred to as the parent-node. For the mapping against the parent-node to be valid, the V-Node must be valid for all child-nodes. This check needs to be performed recursively on all sub-nodes belonging to the relevant child-nodes. All *V-Nodes* created in this step should for readability be numbered with the prefix *'V'* (E.g. $V1, V2, V3, \ldots, Vn$, where $n \in \mathbb{N}$). An example of the use of *V-Nodes* is depicted in Figure 5.4.

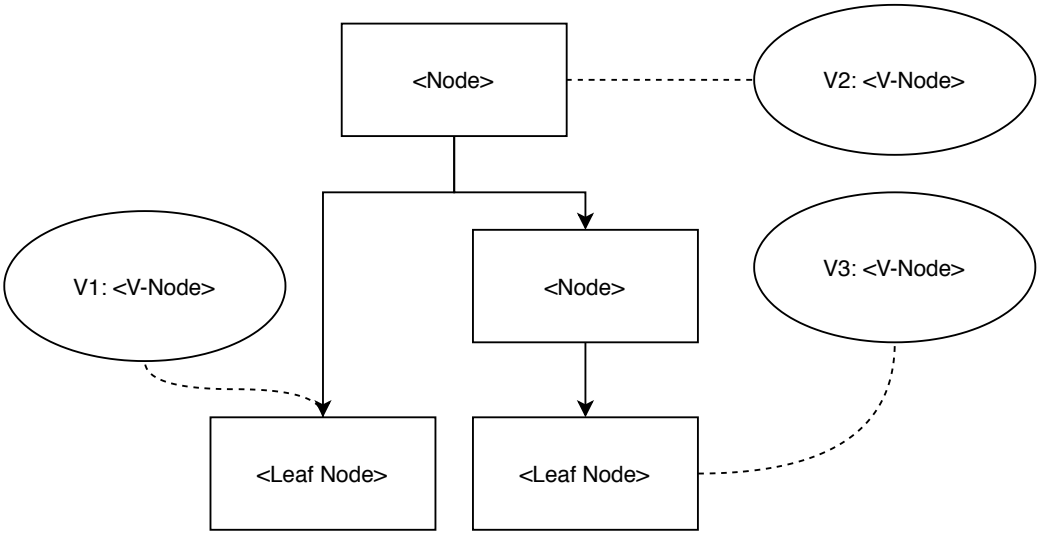

Figure 5.4: Figure depicting an example of the graph created in Section 5.2 with the V-Nodes described in Section 5.4

.

During the mapping process, the *Security Analyst* is expected to use information relevant to the component and technologies used from the different bodies of knowledge suitable to the system. Use of the different bodies of knowledge is required to, for the *Security Analyst* to efficiently use the examinations resources, reflect the overall focus of the testing, and the examinations end-goal. It is highly unlikely for the *Security Analyst* to use all available resources from the different bodies of knowledge, requiring the *Security Analyst* to have some pre-existing knowledge about the use and content of the different external resources.

Using the information described in the *V-Nodes*, the *Security Analyst* matches the possible vulnerabilities with existing information from the different relevant *Bodies of Knowledge*. The *Security Analyst* start with the more generic sources, being either one of the *OWASP*[9, 22] projects, or using Enisa's IOT Tool[25]. Both these resources contain references to either CVE/CWE[14, 15], documents released by other standards organizations, or information published by Enisa[10] and SANS[11]. The *Security Analyst* is expected to add any new information deemed relevant to the examination to the created graph.

The decision of where to start with regards to the different *«Bodies of Knowledge»* is largely dependent on the type of component or system being analyzed. The original *«OWASP Top 10»* project can be relevant in the setting of critical infrastructure, but it is designed around the challenges of web applications[9], and is thus not necessarily fitting for the challenges located in the domain of critical infrastructure and ICS in all settings. The approach is hence designed to take advantage of the more domain specific *«OWASP Top 10 IoT»*[22]. It does not contain external

references but is able to better direct the examination by using the ten listed weaknesses as a starting point. The ten points assist the *Security Analyst* in the design of possible unwanted incidents and to better utilize Enisa's *«Good Practices for IoT and Smart Infrastructures Tool»*.

The described main categories (*«Baseline Security IoT»* and *«Industry 4.0»* presented in Section 4.1.5) are used to assist *Security Analyst* by helping identifying the possible challenges faced by the distinct components or the complex system. This is achieved by determining and searching for likely exposed components and services used by the system and utilizing the tool for gathering information relevant to both the components and specific aspects of the collective system. Information from OWASP [22] and other relevant *Bodies of Knowledge* are utilized by the *Security Analyst* when assessing the information and documents presented by the tool[25].

The gathered information is used by the *Security Analyst* to create a version of the graph where all weaknesses described earlier are replaced with possible mitigations and references to the relevant bodies of knowledge where a component is likely to be vulnerable. These *«answers» or «A-Nodes»* are numbered with the prefix 'A' (E.g. $A1.1, A1.2, A2.1, \ldots, Ax.n$, where $x, n \in \mathbb{N}$) corresponding to the statements from the previous step. In the event that a single statement is split up into several *«A-nodes»* (E.g. if the statement is linked to a category but is moved to a lower level in step 5), these are numbered $A1.1.1, A1.1.2, A1.2.1, \ldots, An.x.y$, where $n, x, y \in \mathbb{N}$. An example of how this is intended to be presented is depicted in Figure 5.5.

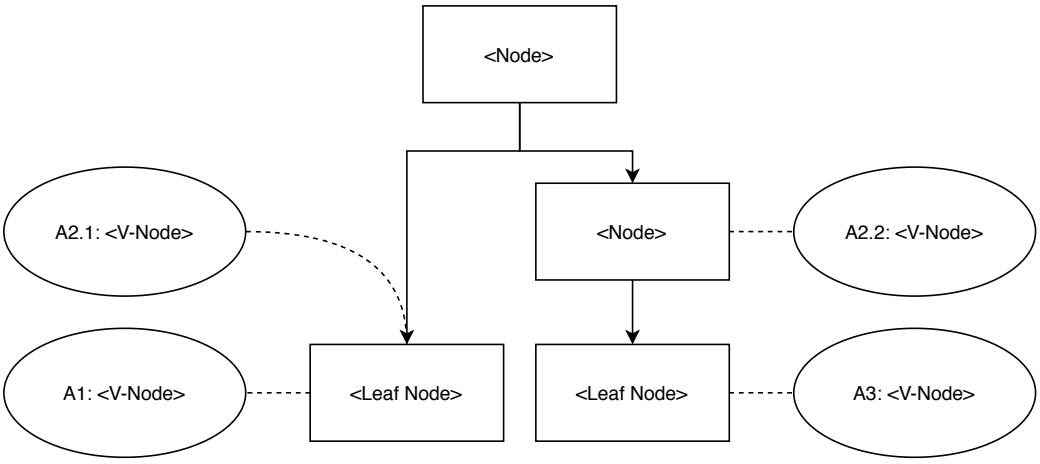

Figure 5.5: Figure depicting an example of Figure 5.4 where vulnerabilities have been replaced with possible mitigations. *V2* from Figure 5.4 is decomposed into *A2.1* and *A2.2*.

If the mitigation descriptions are too extensive for the diagram, they should instead be replaced with a short description and a reference to

a separate document containing the necessary information about the mitigation, and some basic information about how it can be implemented.

It is important to note that information from the resources listed should be used in assistance with the knowledge from both the *Security Analyst* and the *Domain Experts* to develop the tests to fit the use case along with the structure of the system. Using the external resources while looking for possible weaknesses allows the *Security Analyst* to benefit from valuable insight and known challenges within an IoT and Industry 4.0 system from recognized organizations in the field of cybersecurity.

The *Domain Experts* task in this part of the examination is to assist the *Security Analyst* with insight into different historical challenges, or other challenges that either are unique, or notably demanding for *Critical Infrastructure* and *Smart Grid* systems. If there are any physical or other limitations that might hinder the testing process, the *Domain Experts* are expected to work together with the *Security Analyst* for possible solutions. The *Domain Experts* are in addition expected to share knowledge about current trends that might affect the vulnerability of the relevant system or information about particularly critical segments of the system as several observers might reduce bias and selective perception.

5.5 Test the vulnerable components

It is in this stage of the approach important that the *Security Analyst* defines what constitutes a successful testing phase, as testing every component for all possible vulnerability is in most cases unfeasible as a result of constraints on either time or resources. This phase of the approach contains three sub-steps as shown in Figure 5.6.

1. *Test Design & Creation*
   The *Security Analyst* must design test-cases around the possible vulnerabilities and weaknesses described in the V-Nodes from Section 5.4

2. *Testing of Components*
   The *Security Analyst* must test the relevant components.

   (a) If the designed tests are shown to be unsatisfactory, the *Security Analyst* must revisit the design phase.

   (b) If no components are shown to be vulnerable after one or several iterations, the testing phase is complete.

3. *Mitigation of Vulnerabilities*
   Any discovered vulnerabilities must be mitigated, and then tested by revisiting the «Test Design & Creation»-step of the process.

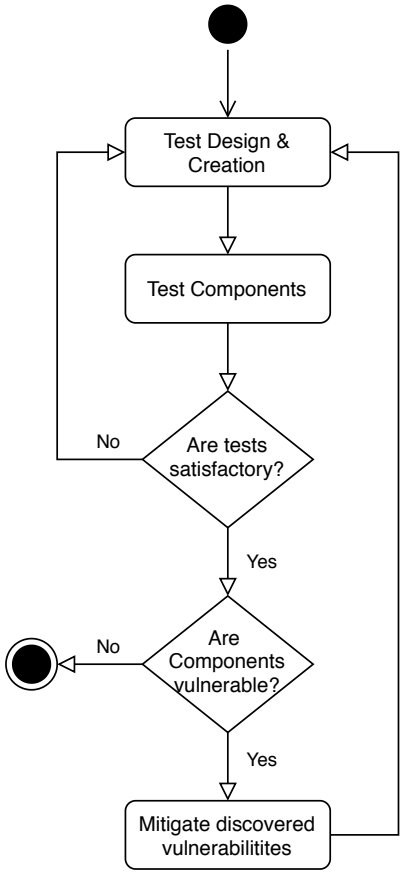

Figure 5.6: Figure describing step 5 of the approach described in Section 5.5 and presented in Figure 5.1.

The test-cases designed in this step are expected to be named by referencing the original vulnerability described in the previous step.

With the assistance of external resources and tools referenced in the previous step, together with the information discovered and collected, the *Security Analyst* is expected to create a set of tests covering the relevant weaknesses. The different tests must be created with a mixed scope, ensuring that the components and functions are not exclusively tested independently outside of the interconnected system. It is expected that the *Security Analyst* uses tests covering different abstraction levels (*Component level, System level etc.*) of the system, with a combination of the different testing levels described in Chapter 4.2.6. This is done in an effort to prevent the components . The *Security Analyst* is additionally expected to use tests from the different external resources gathered in the previous step, and use tests designed on past experiences and information from these sources in an effort to get a varied background on the tests.

The use of external resources to obtain information about steps for testing and mitigation is intended to be a way for the *Security Analyst* to broaden both the scope and depth of the following security testing

of the system. *Enisa*'s IoT tool[25] can assist in the development of tests to improve the coverage and efficiency of the designed tests by acting as a link between the possible vulnerabilities discovered by the *Security Analyst* in the earlier steps, and relevant documents from various *Bodies of Knowledge*. The external tools are expected to be used as a supplementary, increasing test coverage by improving the quality of self-designed tests or adding to the total number of tests if the currents tests are inadequate, they are not intended to be used separately.

The created tests are then used by the *Security Analyst* to test the system. This testing should be done with a focus on the designed testing goals and based on the information contained in the created graphs. During the testing of the system, the *Security Analyst* uses the templates described in Chapter 5.6. The created graphs are expected to give the *Security Analyst* a picture of how the different components affect e.g. a *Smart Grid* with regards to weaknesses, and which vulnerabilities might cover several components. The inheritance-requirement described in earlier steps is still active, and any inconsistencies with regards to this requirement must be amended before the *Security Analyst* can begin the testing.

The *Security Analyst* creates an instantiated version of the graph containing the possible mitigations, with only those that are still relevant to the system. This graph is expected to contain mitigations for the vulnerabilities that have been proven (through testing) to exist within the system. The relevant mitigations found in this graph must be connected to the relevant vulnerable components or categories in accordance with the inheritance principle described in section 5.2. The *Security Analyst* is expected to make an effort in decomposing the relevant nodes to better assist with mitigating the vulnerabilities in the cases where either:

1. Not all listed references are relevant to the component.
2. The different components require significantly different solutions for mitigation.
3. Mitigations not specific enough.

5.6 Test-templates and requirements

The test-phase documentation is divided into two distinct document-types: the tests, and the overall test-plan. The *«test plan»* (Table 5.2) is a collection of different tests, and the relevant documentation while the *«test documentation»* (Table 5.1) contain information about the specific tests and their scope. The content described below includes both optional *(denoted with a '*' in Table 5.2 and Table 5.1)* and mandatory information. The *Security Analyst* is expected to fill out all listed fields relevant to the

examination. The documentation described here is intended to be used throughout the testing phase, gradually filling in information as it becomes available. The «test plan» is intended to act as a parent document for the entire examination, and the *Security Analyst* is thus expected to only have a single document for each system. The *«test documentation* is based around a single either a single test, or a group of tests relevant to the same possible weakness, and a single examination should thus consist of several test documents.

It is important that the designed tests are limited in number and expected time expenditure. Rex et al.[34] describes the need for a balance between the amount of work, and the scope of the testing, as it is impossible to reach a 100% defect detection effectiveness on larger projects with a reasonable resource expenditure[34].

The *Security Analyst* is required to make an effort to design tests that check the same aspect of the system from different angles. Using *Data Source Triangulation* described in Chapter 4.3.4, the *Security Analyst* can check both the consistency and generality of the results. If possible, several *Security Analysts* are included in the process to get a broader background and perspective of the data collected and the system tested in accordance with *Investigator Triangulation* explained in Chapter 4.3.2.

- *Positive Behavior*: User tries to log into the application using username and password
- *Negative Behavior*: User tries to log into the application using SQL-injection in the username-field.

In order to efficiently utilize the allocated resource, the *Security Analyst* is required to design test-cases in such a way that, where possible, a single test-case is either applicable to several components or including several test-conditions. This does however create issues when testing a system for negative behavior, as Rex et al.[34] describes the issue of how a system handles errors and unexpected inputs. The negative behavior might induce unexpected behavior in the system preventing e.g. all but one of the test-conditions to be checked or preventing an unknown number of test-conditions to be checked.

Test documentation content	
The system and the components	• The testing is performed on multiple components, or the system as whole, depending on the type of vulnerability
The time and date	• This is in the form of both a start and end date, and a start and end time to accurately measure the time spent on a specific test.
Test-specific expected time and man-hours used (*)	• This must contain both the number of hours expected to be used on the project, and the number of people involved.
The Security Analyst	• The name of all includes Security Analysts, everyone listed here is also included in the personnel-list described in the test-plan
The test-objective	• The purpose of this test within the bigger test-plan. This must be consistent with the "statements" described in step 4 of the approach.

Table 5.1 continued from previous page	
Test-specific assumptions and scope	• All assumptions relevant to the test not listed in the test-plan • Clearly defined test scope with information about what is and what is not included in the test. These limits can be the separation between the physical and cyber-realm or defining borders in the system where e.g. communication from component A to B is defined as within the scope, while communication from A to C is defined as outside of scope and is covered elsewhere.
Detailed explanation and schedule of the test (*)	• Relevant references for tests based on pre-existing work, e.g. *«The OWASP Testing Guide»*. • It is required to include detailed information to allow the reproduction of results by a person not previously involved in the testing of the relevant system.

Table 5.1 continued from previous page	
Test-input/action, Expected test-result, Actual test-result	• This is required to be described in the form of: – Action: *<This is the specific input/setting/action the system is tested with>* – Expected behavior: *<How the system is expected to act in a specific setting>* – Actual behavior: *<How the system acted in a specific setting>*
A test conclusion	• A description of any difference between the expected and actual behavior. Is it necessary to either create new tests or revise the existing? • Information about ease-of-exploitation, any assumptions from the *Security Analyst* with regards to severity, and an example of a possible exploit. This example is described in the form of a CORAS[33] or a misuse case diagram[38]. • Information about relevant mitigations where available for test from external resources,

Table 5.1: Table describing the expected content of the testing documentation described in Section 5.6. *Optional

Test plan content	
The tested system and components	• This is required to be consistent with the scope and information defined in the first step of the developed approach
The start and end time/date for the testing period (*)	• The total time and manhours expected to be required for planning, documenting, and testing the system
The overall testing objective	• This includes all the *«Statements»* described in step 4 of the approach
The name and background off all included *Security Analysts*	• It is necessary with a complete list of everyone involved and a summary of their background to ensure efficient resource use
General assumptions (* Depending on the system, there might not be any systemwide assumptions)	• All assumptions valid for the system as a whole is listed here, this includes all relevant assumptions made in step 1 of the developed approach.

Table 5.2 continued from previous page	
A list of documentation for all planned tests within the test-plan	• This list is required to be presented in a way that an actor not involved with the testing is able to navigate the test-plan and the referenced tests. • The list can be ordered in two ways: 1. Sorted on the affected components, which shows which components are affected by the highest number of bugs and allows prioritizing based on this attribute. 2. Sorted on the *«Statements»*, which shows which vulnerabilities are widespread, and allows prioritizing based on this attribute.
Information about which *Security Analyst* is responsible for the different tests within the test-plan	• Including references to the relevant tests.
A summary with regards to the overall test-results	• Including any information about the prioritizing of some test results over others with regards mitigations.

Table 5.2: Table describing the expected content of the testing plan described in Section 5.6. *Optional

Chapter 6

Trial of the approach

6.1 Approach illustrated in the setting of a Smart Grid with self-healing properties

We use [8] as an inspiration in order to exemplify usage of the approach proposed. The trial presented in this Chapter, as well as the results obtained, are entirely fictitious and only intended to illustrate the application of our approach. There is otherwise no relationship between the trial of the approach (context/results) presented in this chapter and [8].

6.1.1 Establish context

In the setting of a smart grid with self-healing properties we have described the context of the examination, limiting the scope to the components described in [8]. We defined the goal of this examination to be the reliability of the Self-Healing grid, focusing the tests on vulnerabilities that negatively affect the availability of power for the customer, and vulnerabilities that allow an external actor insight into the inner workings of the system. In this examination, we have the role of the *Security Analyst*, with the paper by Omerovic et al. [8] in the role of a *Domain Expert*. We have assumed the *Security Analyst* to have a background in software engineering and cyber security, with some domain knowledge about critical infrastructure, sensor technologies, Industrial Control Systems (ICS) and Smart Grid systems. The expected timeframe for this examination, including testing is one week, with a single security analyst. These assumptions created a baseline for our expectations with regards to both the scope and size of the examination and the following testing. As we analyzed a complete system, we examined all components within the chain of communication between the sensors and actuators, and the *SCADA* control system. This included the *Remotely Controlled Switches*, *Fault Indicators*, *Controllers/IEDs*, *Substation RTU*, the *Distributed Management System*, and the *SCADA*

system. We described these assumptions and the different components in the form of several lists for readability.

Components within scope:

1. *Remotely Controlled Switches* (RCS)
2. *Fault Indicators* (FI)
3. *Intelligent Electronic Device* (IED, Controller)
4. *Substation Remote Terminal Unit* (RTU)
5. *Distribution Management System* (DMS)
6. *Supervisory Control And Data Acquisition* (SCADA)

From these Components, we have described the categories and the groupings based on common attributes and logical placement within the grid based on Figure 6.1. As both the *FI* and the *RCS* are described as a part of a Self-healing node, and there are two types of self-healing nodes we have created the categories; *Master Node*, *Slave Node*, and *Self-healing Node* and group them according to the relationship between themselves. E.g. a *Fault Indicator* is a *Slave Node* which is a *Self-Healing Node*. We have differentiated between the *Slave Node* and the *Master Node* based on their difference in responsibility. The *Master Node* acts as a gate between the *Substation* and the node-network, as well as managing the node-network.

1. *Master Node:* Contains the *RCS* and *FI*
2. *Slave Node:* Contains the *RCS* and *FI*
3. *Self-Healing Node:* Contains the *Master* and *Slave Node*
4. *Substation:* Contains the *Master Node*, *IED*, and *RTU*
5. *Control Systems:* Contains the *RTU*, *DMS*, and *SCADA*-systems
6. *Self-Healing grid:* Contains the *Self-healing Nodes*, *Substation*, and *Control Systems*

We have defined the scope of the examination and made assumptions based on who is performing the examination, we have made an effort to remove any vagueness with regards to the relationships between the components. We have explained the reasoning behind differentiating between the *Slave node* and *Master node*. Why are they listed as separate components when they have the same relationship with both the *RCS*, *FI*, and the *Self-Healing Node*?

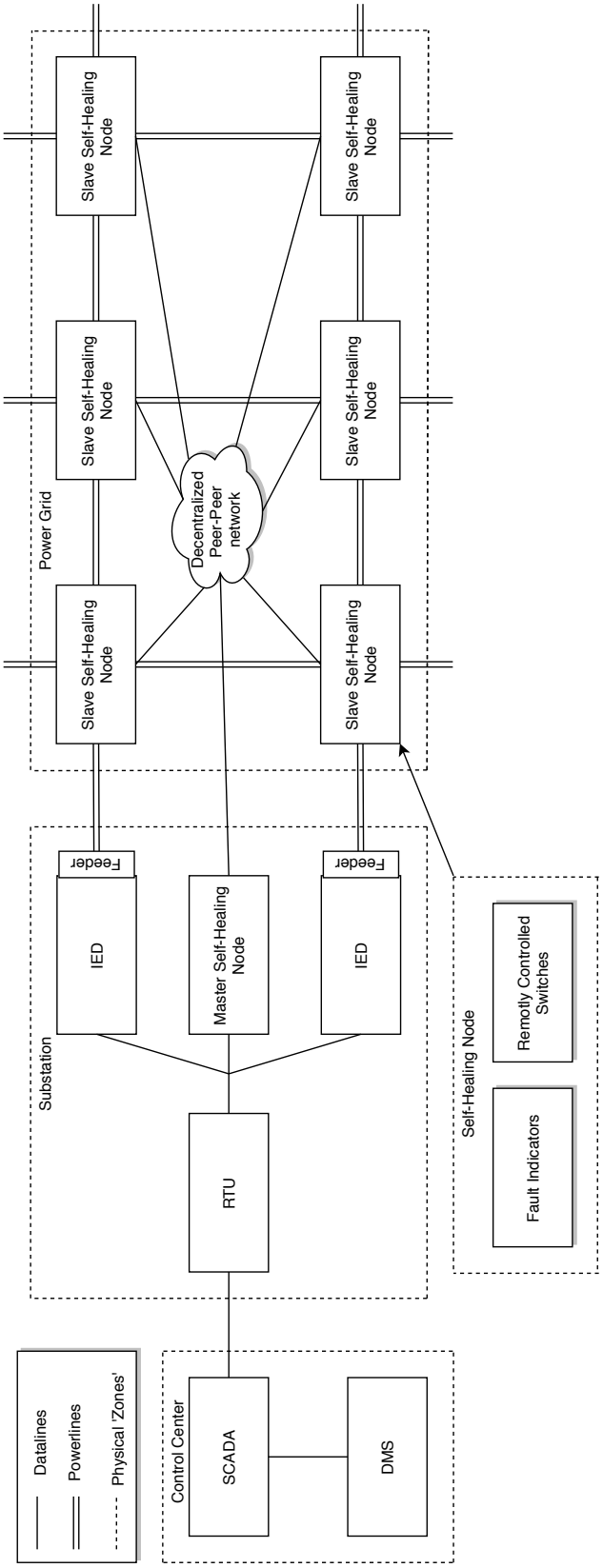

Figure 6.1: Figure depicting the relationship between the components described by Omerovic et al.[8]

Assumptions:

1. The scope of the examination is limited to the components described in Omerovic et al. in [8] and Figure 6.1.
2. The Security Analyst has a background in software engineering and cyber security with some domain knowledge about ICS, Sensor- and Smart Grid-technologies.
3. The *Slave Node* and *Master Node* are identical, apart from the *Master Node* receiving direct communication from the *Substation* and acting as an intermediary between the network of *SH-Nodes* and the rest of the system.
4. The *SH-Nodes* communicates using a local wireless peer-to-peer network.
5. The *RTU* and the *SCADA*-system communicate through a VPN-tunnel on an IP-based network.
6. The *Feeder* described in Figure 6.1 is being directly managed by the IED and has no significant cyber footprint by itself.

6.1.2 Map components within the scope

By using the *«rules»* describing the component and category relationships, we have designed a non-cyclical directed graph visualizing the relationships between all the listed categories and components. Every category and component defined in the *«rules»* result in a single node in the graph, and each relationship from a category to a sub-category or from a category or sub-category to a component is depicted as an arrow moving towards either the sub-category or the component, where the component is placed on the lowest level of the graph.

The graph is built up of several abstraction-levels, where the components, in the form of leaf-nodes, are at level 0, with a different depth of the various sub-trees, we group the different levels depending on their level of abstraction. The depth of the sub-tree containing the *Fault Indicators* is three (*SH-Node -> Slave Node -> FI*), while the depth of both the sub-trees containing the *Substation RTU* is two (*Substation -> Substation RTU*).This difference has an impact of the structure and internal grouping of the tree, we have thus made decisions with regards to which nodes are positioned on the various levels. As all components belong to the same abstraction-level the decision has been made on the location of the components parent-nodes.

- The grouping of the nodes within the *Self-Healing nodes* subtree is fairly simple, both the *Master Node* and *Slave Node* have the same

parent and children nodes with only minor differences described in the assumptions in section 6.1.1. We have positioned the *FI* and *RCS* at the same abstraction level following the rule described that all components are positioned at level 0. The similarities in the relationship between the *Master Node* and *Slave Node* and their parents and children require them being placed on a similar abstraction level. The sub-tree is presented in Figure 6.2.

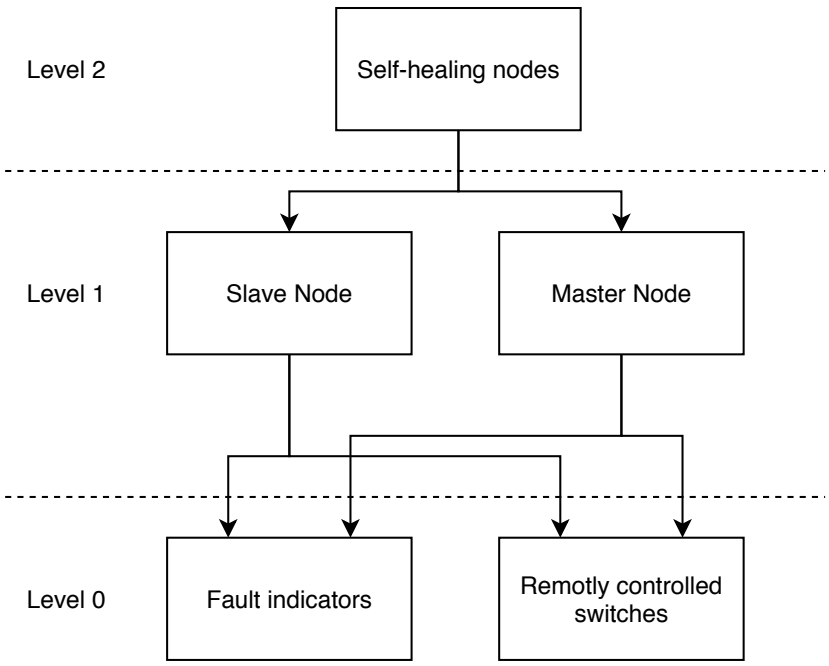

Figure 6.2: The mapping and placement of the categories belonging to the Self-Healing Node sub-tree

- The grouping of the nodes within the *Substation* sub-tree is built up of three levels. The sub-tree contains both the *Master node* sub-tree, and the leaf-nodes *IED* and *Substation RTU*, and with regards to the rule that all components are on the same abstraction level, we have designed the *Substation* sub-tree with all components on level 0, the *Master Node* on level 1 matching its location in Figure 6.2, and the *Substation* on level 2, as the *Master Node* is not allowed to inherit from a category on a similar or lower level. The sub-tree containing the *Substation* is presented in Figure 6.3.

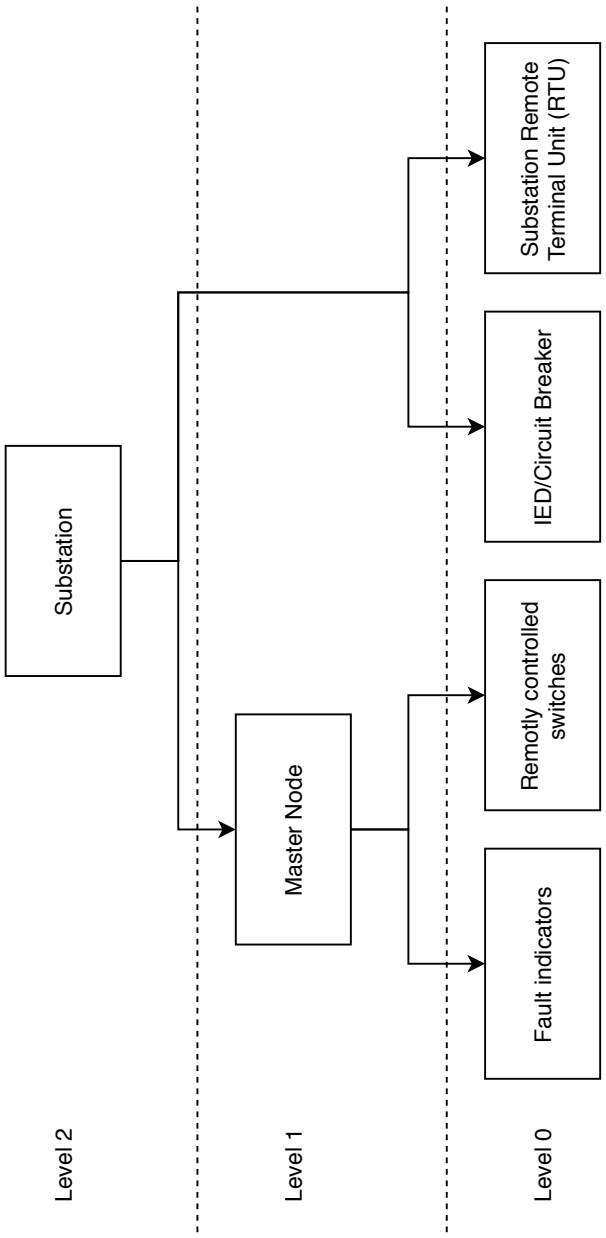

Figure 6.3: The mapping and placement of the categories belonging to the substation sub-tree

- The grouping of the nodes within the *Control Systems* sub-tree is comprised of two different levels, the root-node *Control Systems*, and the different components; *Substation RTU*, *DMS*, and *Scada*. The abstraction-level of *Control System* is of similar scale to the *Self-Healing Nodes* and *Substation*, and is thus not of equal level as the *Master Node* and *Slave Node*. We have placed the *Control System* root node at the same abstraction level as the *Substation* and *Self-Healing node*, this requires a gap between the *Control Systems* and the various components, as they are both required to match the

abstraction level of the previously described components belonging to the *Substation* and *Self-Healing nodes* sub-trees, and the *RTU* described in Figure 6.3 is not allowed to exist on separate abstraction levels. The sub-tree containing the *Control Systems* is presented in Figure 6.4.

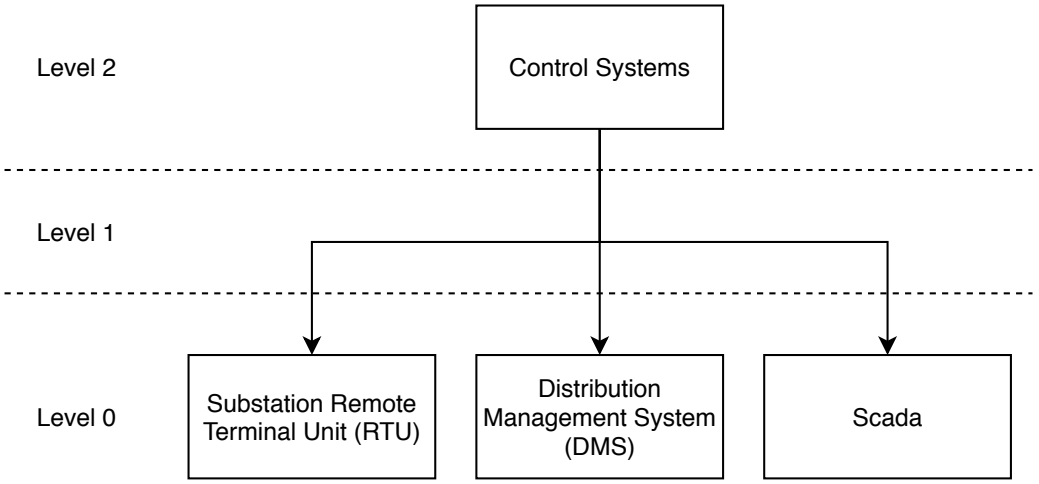

Figure 6.4: The mapping and placement of the categories belonging to the Control Systems sub-tree

The mapping and the placement of the different categories, sub-categories and components is presented in Figure 6.5.

6.1.3 Validate the correctness and consistency of the examination

In this step we validate the created graph with regards to correctness with the real-world, and to the assumptions and design choices made in previous steps. As Figure 6.5, described in step 2 is strictly a non-cyclical directed graph with no connections in the direction of either a node on the same level, or a node at a higher level, the design does not contain any cycles and fulfills the syntactical requirements. Every component described in step 1 has been depicted, and the grouping is consistent with the specification described in step 1 and 2.

With the graph being an accurate representation of the data described in the previous steps, we have designed clear rules with regards to the scope and design choices we have made. Including both the validity of the node groupings and the design choices regarding the positioning and the abstraction levels in the semi-hierarchical structure.

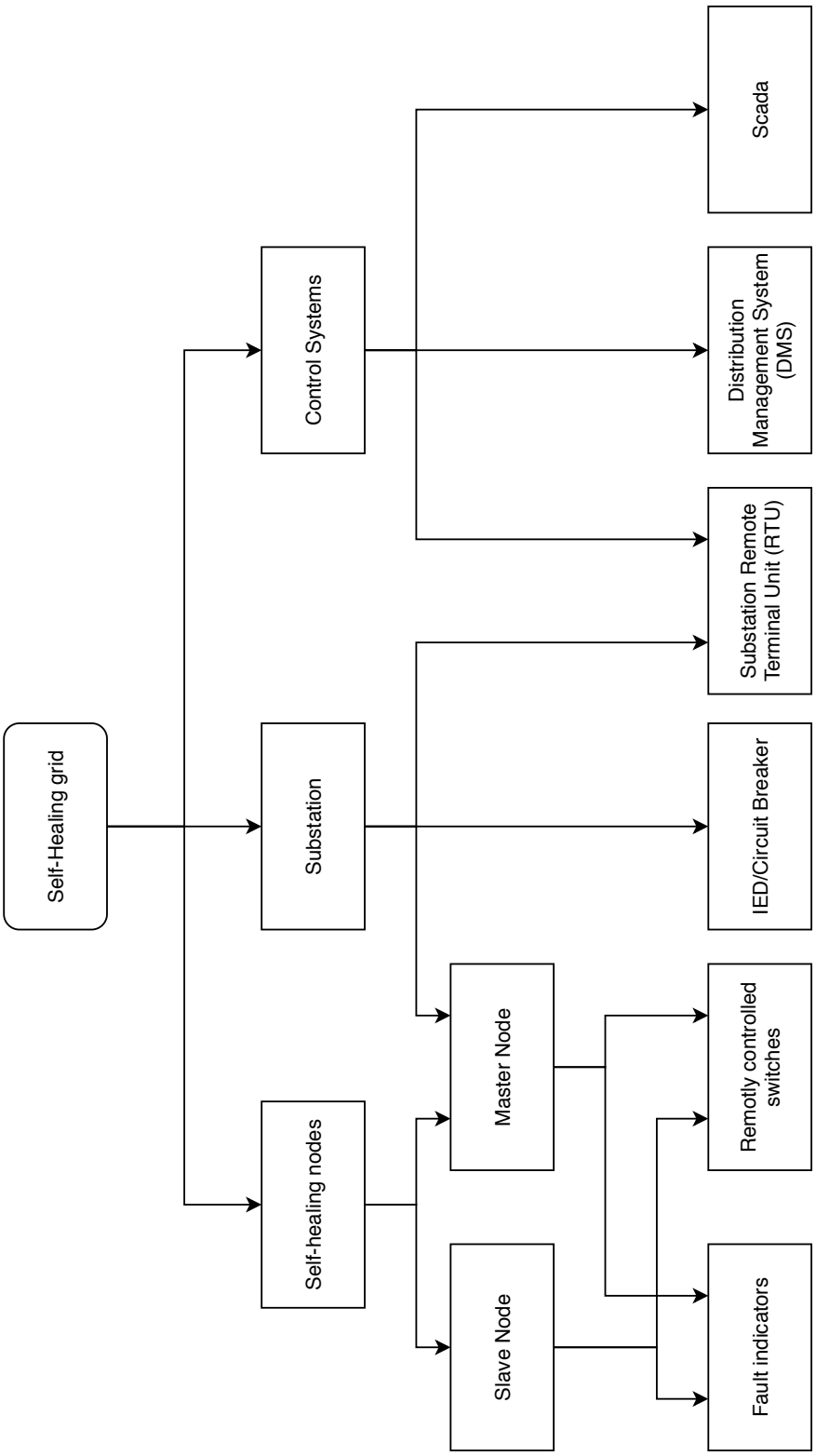

Figure 6.5: Figure depicting the non-instantiated graph with regards to a Smart Grid with self-healing capabilities

Within scope:

1. The *Self-Healing nodes* are within scope of the examination because they both collect information about possible faults and relays this information to the *Control Systems* via the *RTU*.

2. The *IED* is within scope of the examination because it in this setting acts as a controller for circuit breakers which gives it the ability to cut off power.

3. The *RTU* is within scope of the examination because it is required for communication between the *Control Systems* and the *Substation*-connected equipment.

4. The *SCADA* and *DMS* are within scope of the examination as they are used by the *DSO* to monitor and control the grid based on data collected during operations from the *RTU*.

Outside of Scope:

1. The Feeder described in Figure 6.1 is outside of the scope of this examination as it is a physical power line with no cyber footprint, instead being connected to the *IED*.

Groupings:

1. The *Slave Node* and *Master Node* are a part of *Self-Healing Node* because they are assumed to be identical outside of the communication between the *Master Node* and the *RTU*, requiring the differentiation.

2. The *RCS* and *FI* are a part of both *Slave Node* and *Master Node* because we earlier assumed these categories to be identical.

3. The *Master Node*, *IED*, and *RTU* are a part of the Substation because both the *Master Node* and *IED* communicates directly with the *RTU* and is contained within the same «physical»-zone depicted as an inter-connected network within the *Substation* shown in Figure 6.1.

4. The *RTU*, *DMS*, and *SCADA* are parts of the *Control Systems* because the *DMS* and *SCADA* both collect data and monitors the system by use of the *RTU*.

6.1.4 Map possible weaknesses and vulnerabilities to all relevant categories

Using the CORAS language we have described several unwanted incidents, threat scenarios, and vulnerabilities around the assets Confidentiality, Integrity, Availability, and Non-Repudiation. We have not differentiated the threat actors other than it being either the work of a malicious actor, or a non-human threat, and we have not made any assumptions with regards to the frequency and likelihood of these events.

- *Confidentiality*: A loss of data confidentiality can lead to loss of either system or user data
- *Integrity*: A loss of data integrity through faults, discrepancy between expected data and actual data, or malicious action can mislead either the operator or a component into making improper actions in response to changes.
- *Availability*: A loss of system availability can lead to a power outages or loss of control of critical systems.
- *Non-repudiation*: A change or action done by an unknown actor or impersonator in the name of a trusted party (e.g. a component or the operator) can have unwanted consequences blamed on the wrong party.

We have described generic incidents which can negatively affect these assets in accordance with our testing goals using information from the *OWASP Top 10 IoT-project*[22], while it is fairly generic and is not directly intended for use with industrial IoT, several of the listed common security issues are relevant to a Self-healing Smart Grid system. We have created a list containing these security issues combined with a justification for why we have deemed them relevant for a Smart Grid System with Self-Healing properties:

- 2. Insecure Network services
  - Use of existing phone or IP-network for remote management increase the attack surface of a system, allowing previously protected equipment to be reachable.
- 7. Insecure Data Transfer and Storage
  - The distributed nature of a power grid requires communications between geographically dispersed components which can contain sensitive information.
- 8. Lack of device management

  - Placement of equipment outside of a controlled environment, where physical access can be a challenge requires a significant level of precise device management.

- 9. Insecure default settings
  - Mistakes or misunderstandings during component installation can lead to vulnerable systems.

We used Enisa's IoT Tool [25] with a basis on the listed security issues. This was conducted by selecting the *«Security Measures»* similar in name or intent to those listed by OWASP[22], (*E.g. «Cryptography», found in the tool is relevant to «Insecure Data Transfer and Storage»*) we have presented the aggregated information in the form of a list containing the *«Security Measure»*, the selected *«Best Practice»*, and the referenced source:

2. Insecure Network services
   1. *«Implement a DDoS-resistant and Load-Balancing infrastructure to protect the services against DDoS attacks which can affect the device itself or other devices and/or users on the local network or other networks.» NIST SP 800-53 - SC-5 Denial Of Service Protection[39]*
   2. *«Protocols should be designed to ensure that, if a single device is compromised, it does not affect the whole set, since smart objects are often deployed as sets of identical or almost identical devices.» NIST SP 800-53 - SC-5 Denial Of Service Protection[39]*

7. Insecure Data Transfer and Storage
   1. *«Ensure a proper and effective use of cryptography to protect the confidentiality, authenticity and/or integrity of data and information (including control messages), in transit and in rest. Ensure the proper selection of standard and strong encryption algorithms and strong keys and disable insecure protocols. Verify the robustness of the implementation.» NIST SP 800-53 - SC-13 Cryptographic Protection[39], ISO27001#A10 Cryptography[40]*

8. Lack of device management
   1. *«Ensure minimal level of authentication security for the IoT devices and systems. In a segmented network/system, ensure that authorization only allows for access to a certain segment*

and no other parts of the system.» Identity and Access Management for the Internet of Things - Summary Guidance - IoT Working Group[41]

9. Insecure default settings

    1. *«Enable security by default. Any applicable security features should be enabled by default, and any unused or insecure functionalities should be disabled by default. Strong security controls should be something the consumer has to deliberately disable rather than deliberately enable.» Identity and Access Management for the Internet of Things - Summary Guidance - IoT Working Group[41], ISO27001#A12 Operations Security[40]*

With the use of OWASP[22] and the information gathered from the tool we have described a possible scenario: a malicious actor is able to eavesdrop on the communication between either the *Slave Nodes* or the *Slave Nodes* and the *Master Node* due to lack of, weak, or incorrectly implemented data protection schemes on inter-component communication. This scenario could lead to the loss of confidential system data, which harms the «Confidentiality»-asset. With the distributed nature and compute power limitations of this system, lack of both effective and resource-efficient encryption while at rest and during transmission have a major impact on the security of the system. The distributed nature of the system has created challenges in the responsiveness and requirements for local decision making in the face of both lack of or erroneous communication from other components. These challenges in combination with the four possible security issues listed above and the gathered information, was used to devise a list of possible scenarios damaging our assets, with a special focus on incidents that might do harm to the availability of the grid.

We composed the different scenarios in the natural language in the form:

> <*Vulnerability*> result in <*Threat Scenario*> leading to <*Incident*>, harming <*Asset*>.

This was done to facilitate the use and creation of a CORAS diagram[33] and we grouped them based on the underlying weakness. We have presented these scenarios in the form of a list with an index reference for traceability to the list containing *Security Measures* and *Best Practices* presented in the beginning of this section.

1 *Lack of protection on inter-component communication* result in *Insecure communication allows eavesdropping of communication* leading to *loss of confidential user and system data*, harming *confidentiality*.
(Based on 7.1)

2.1 *Component exposed to the internet* result in *Exposed component allows un-authenticated connection to equipment* leading to *loss of confidential user and system data*, harming *confidentiality*.
(Based on 7.1, 8.1, 9.1)

2.2 *Component exposed to the internet* result in *Exposed component allows un-authenticated connection to equipment* leading to *alteration of system data* and *disruption of system*, harming *confidentiality*, *non-repudiation*, and *availability*.
(Based on 7.1, 8.1, 9.1)

2.3 *Component exposed to the internet* result in *Exposed component allows un-authenticated connection to equipment* and *Malicious commands take s down components*, leading to *a disruption of the system*, harming *availability*.
(Based on 7.1, 8.1, 9.1)

3 *Wrongly configured component* result in *Incorrect FI configuration leads to power shut-off*, leading to *a disruption of the system*, harming *availability*.
(Based on 8.1, 9.1)

4 *Wrongly configured component relationship* result in *Rule uncertainty prevents Slave from accepting commands from Master*, leading to *a disruption of the system*, harming *availability*.
(Based on 2.1, 8.1, 9.1)

5 *Lack of local security functions* result in *Error damages component when unable to receive data from Management*, leading to *a disruption of the system*, harming *availability*.
(Based on 2.1, 9.1)

We have described simplified versions of the listed scenarios with the relevant possible vulnerabilities within the scope of the examination in the form of a CORAS-diagram in Figure 6.6 to present the scenarios and their effect on the different assets.

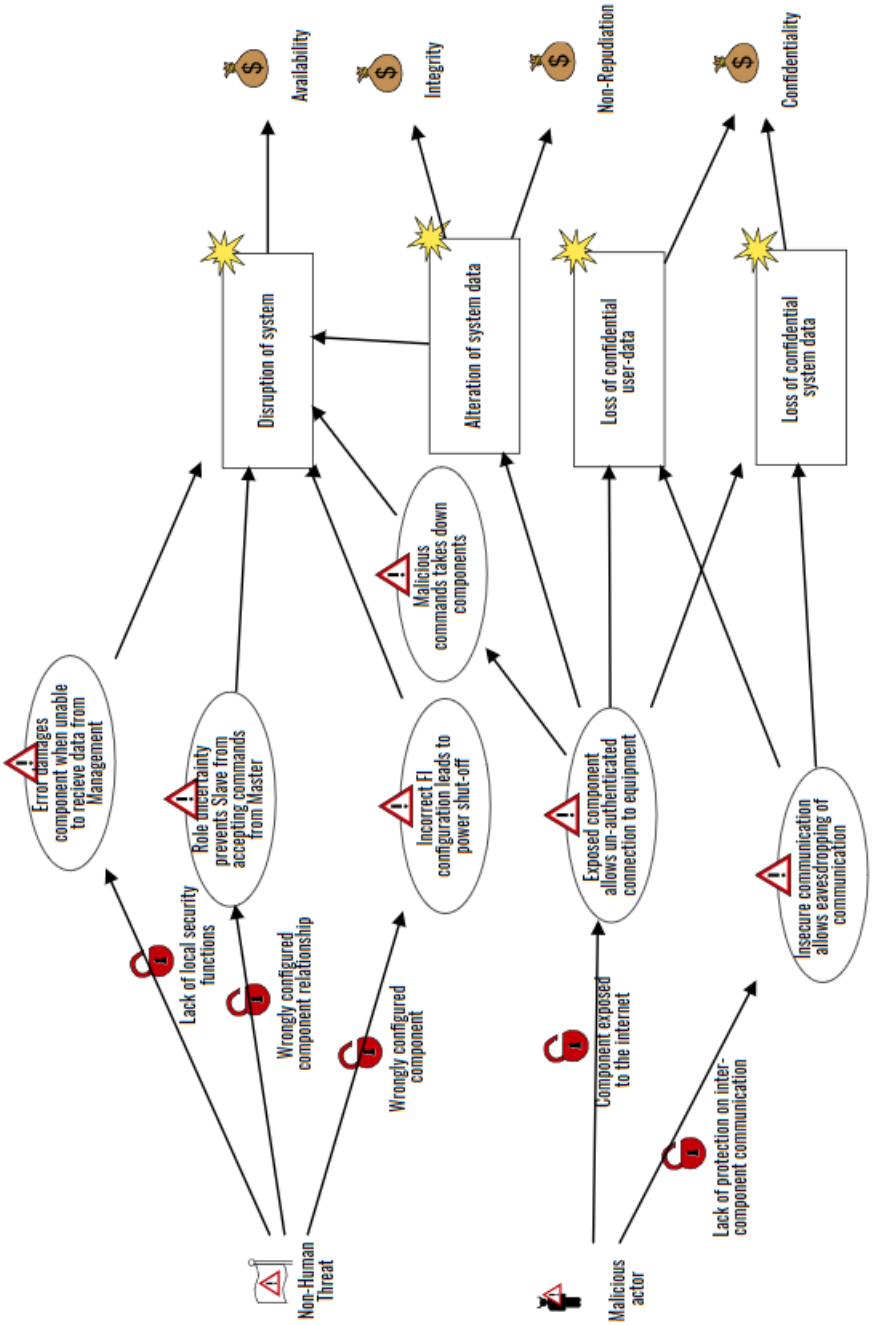

Figure 6.6: Figure describing vulnerabilities relevant to loss of Confidentiality, Integrity, Availability, and Non-Repudiation in the setting of a Smart Grid with self-healing capabilities.

The information about the vulnerabilities we described in Figure 6.6 have been simplified further, and the information has been included in Figure 6.7 with connections to all the relevant components in the form of the V-nodes we described in Section 5.4.

The information about the possible mitigations we have gathered during the examination are depicted in Figure 6.8, a modified version of Figure 6.7 where the V-nodes have been replaced by the A-nodes described in Section 5.4.

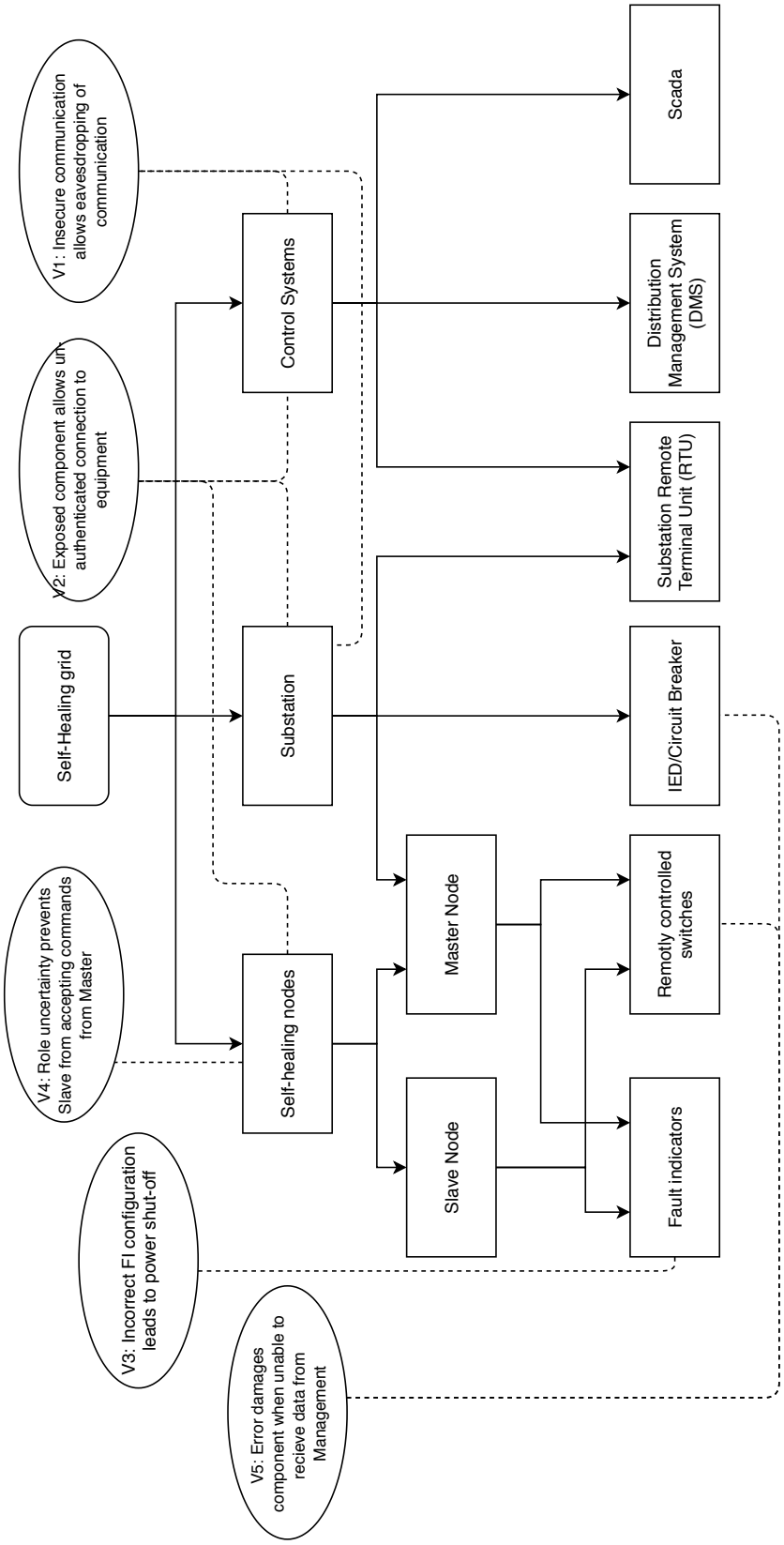

Figure 6.7: A modified version of Figure 6.5 with possible vulnerabilities from Figure 6.6

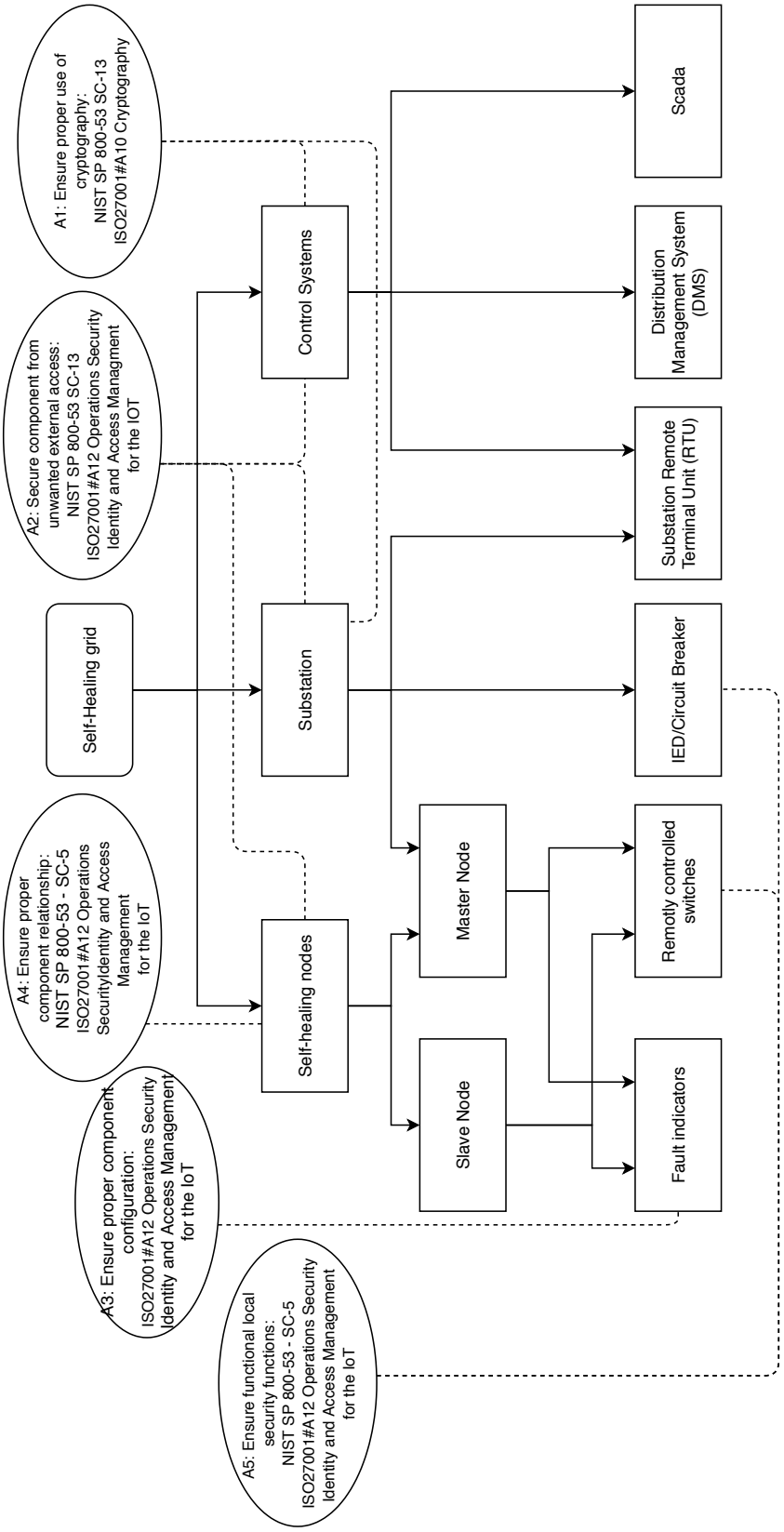

Figure 6.8: A modified version of Figure 6.7 with possible mitigations and the relevant references gathered using the external resources while searching for the weaknesses depicted in Figure 6.7.

6.1.5 Test the vulnerable components

As this trial has been used to support the development of our testing approach, verification of both the tests and information about what constitutes a successful testing phase was outside of our scope. If this was a real-world example, we would here have defined what would be required for the testing phase to be complete. With regards to the tests and their results, we have provided generic test descriptions, and shown how both satisfactory and unsatisfactory test-results are modeled into the various graphs.

Using the information described in the V-nodes of Figure 6.7 along with assumptions to the functions of the different components, e.g. the assumptions regarding communication between *SH-nodes* and communication between the *Substation* and the *Control Systems*, we have expanded the V-nodes, which lead to several tests specific for each connected node.

V1: Lack of protection on inter-component communication can lead to eavesdropping
Relevant to: *Substation, Control Systems*

- V1.1: Test if an actor inside the network is able to eavesdrop on *Substation* or *Control Systems*
- V1.2: Test if an actor outside the network is able to eavesdrop on *Substation* or *Control Systems*
- V1.3: Test if data acquired from eavesdropping on *Substation* or *Control Systems* is understandable

V2: Component exposed to the internet can lead to unauthenticated access
Relevant to: *Self-Healing Nodes, Substation, Control Systems*

- V2.1: Test if *Self-Healing Nodes, Substation, Control Systems* are reachable from outside the network
- V2.2: Test if *Self-Healing Nodes, Substation, Control Systems* accepts data from un-authenticated users or components

V3: Wrongly configured component can lead to erroneous power shut-off
Relevant to: *Fault Indicator*

- V3.1: Test how *Fault Indicator* acts in an erroneous state
- V3.2: Test how *Fault Indicator* acts to fluctuating powerline changes

V4: Wrongly configured component relationship can lead to miscommunication between nodes
Relevant to: *Self-Healing Node*

V4.1: Test if *Slave Node* acts on commands sent from other *Slave Node*

V4.2: Test if *Slave Node* acts on commands sent from something impersonating *Master Node*

V4.3: Test if *Slave Node* will relay command from something impersonating *Master Node*

V4.4: Test if *Master Node* acts on commands sent from *Slave Node*

V4.5: Test if *Master Node* acts on commands sent from something impersonating *Master Node*

V5: Lack of local security functions can lead to damages if unable to receive commands
Relevant to: *RCS, IED*

V5.1: Test how *RCS* and *IED* handles erroneous data when unable to communicate with other Components.

V5.2: Test how *RCS* and *IED* handles regular data when unable to communicate with other Components.

V5.3: Test if *RCS* and *IED* are able to respond and react to data indicating a malfunction when unable to communicate with other Components.

V5.4: Test how *RCS* and *IED* are able to function with unstable communication with other Components

We have assumed the tests *V1.3, V2.2, V3-V5* passed, and that *V1.1, V1.2, V2.1* to some degree demonstrated that the tested components were vulnerable. We described the imagined outcome of the failed tests in the form of a list:

V1.1 Showed that an actor inside the network is able to eavesdrop on communication from *Substation*

V1.2 Showed that an actor outside the network is able to eavesdrop on communication from *Substation*

V2.1 Showed that both *Self-Healing Nodes* and *Control Systems* are reachable from the internet

Using the test results, we have instantiated Figure 6.8, keeping the mitigations relevant to the components proven to be vulnerable, and discarding the rest. This resulted in Figure 6.9

Following the inheritance principle and the decomposing we described in Section 5.5, both of the mitigations from Figure 6.8 now described in Figure 6.9 were lacking in both information and relevance to the proven

vulnerabilities. With the use of the referenced external resources, we decomposed *A1* and *A2*, creating three distinct A-nodes based on the name of the described tests:

> *A1 «Ensure proper use of cryptography»*, relevant to both the vulnerable internal and external traffic discovered by *A1.1* and *A1.2*)
>
> *A2.1 «Secure component from unwanted external access»*, Decomposed into *A2.1.1* and *A2.1.2* due to being to generic
>
> - *A2.1.1 «Ensure only the necessary ports and interfaces are reachable»*
> - *A2.1.2 «Boundary protection Devices to prevent access to internal network from outside»*

We replaced the *A-Nodes* in Figure 6.9 with the decomposed *A-Nodes*, creating a duplicate of *A2.1.2*, following the requirement of similarity described in section 5.5, resulting in Figure 6.10.

The discovered vulnerabilities would in an actual case study be mitigated by following the instructions and guidelines from the external resources found in Figure 6.10 using the iterative process described in Section 5.5.

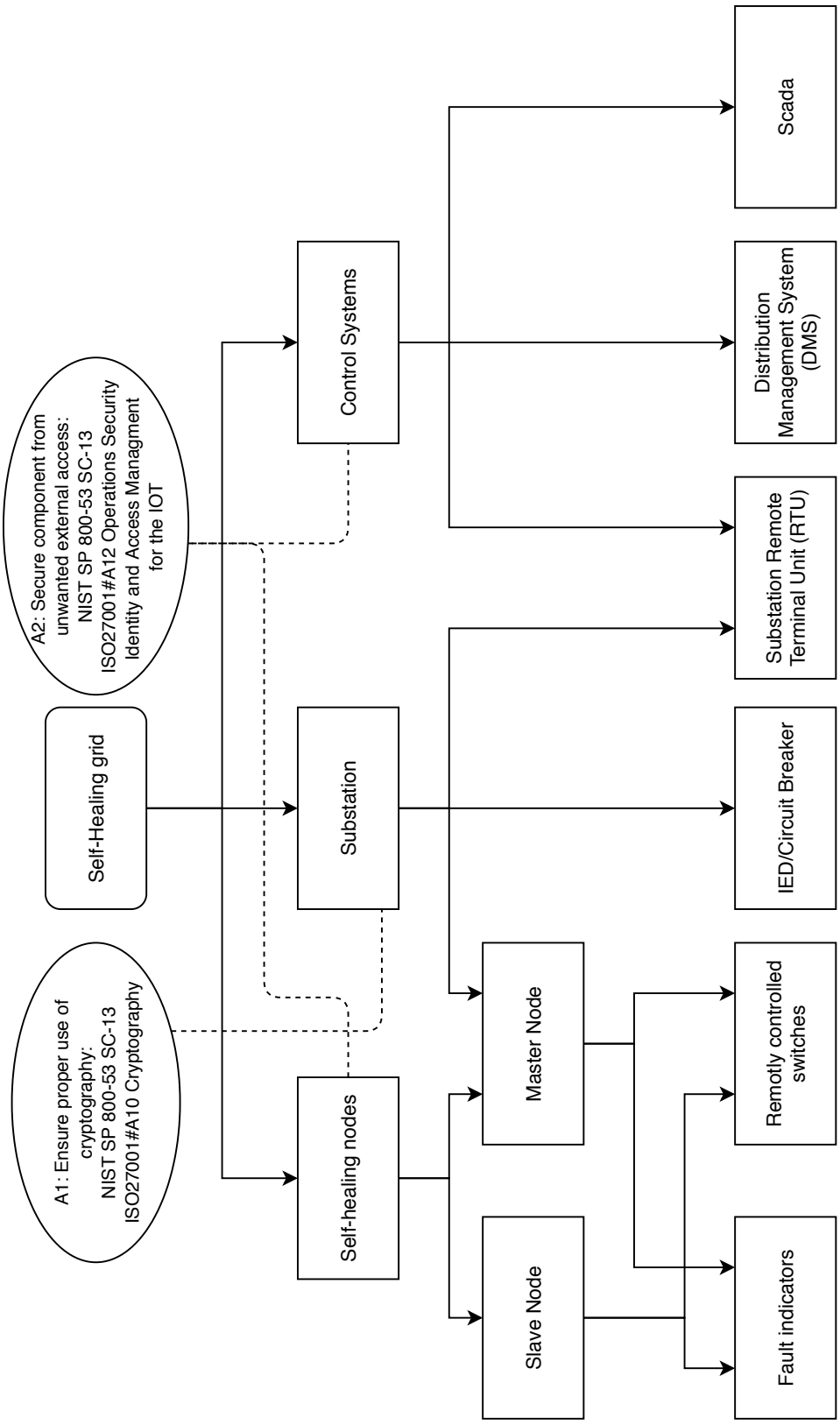

Figure 6.9: Instantiated version of Figure 6.7 containing mitigations for components tested to be vulnerable.

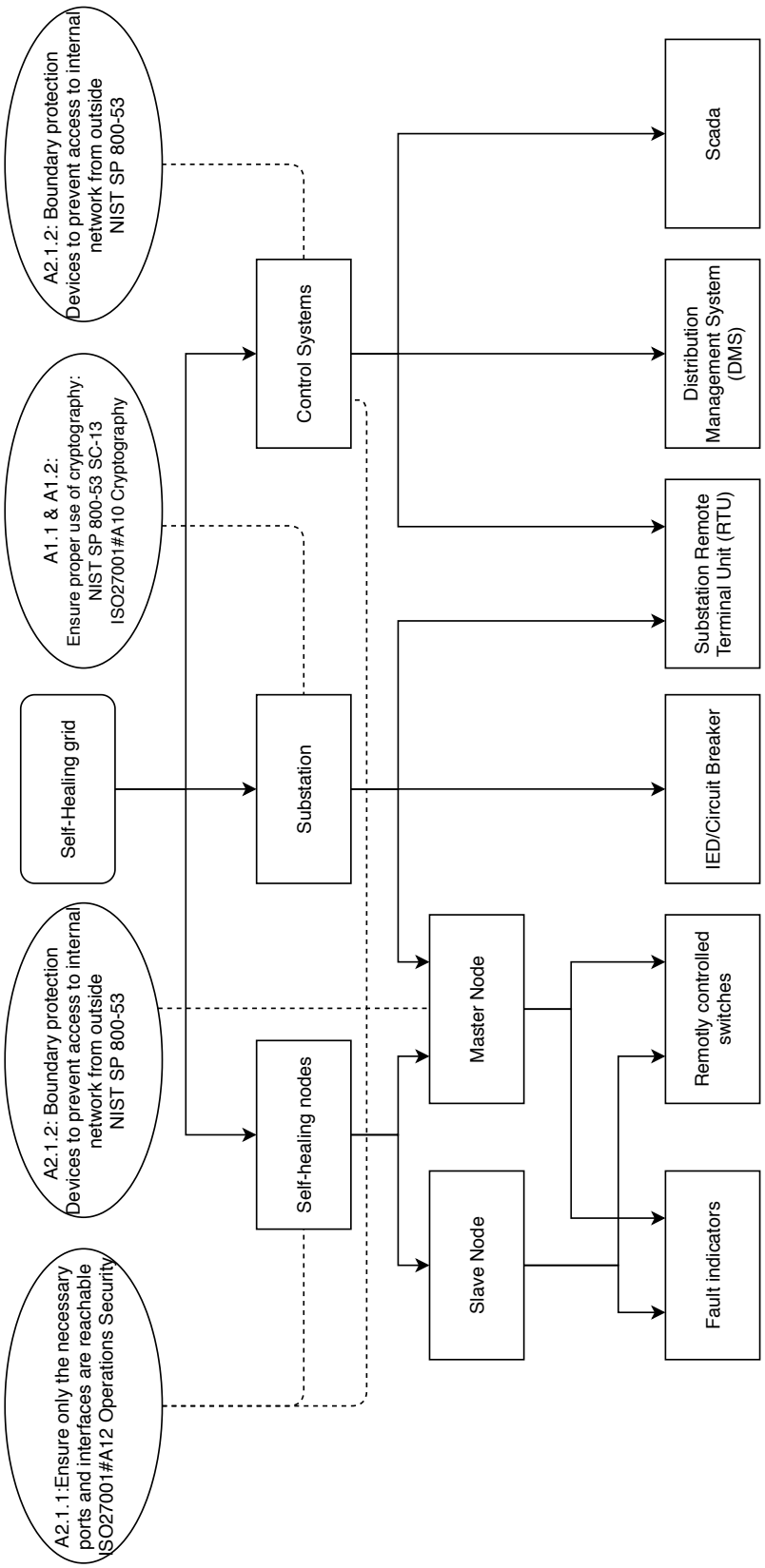

Figure 6.10: Modified version of Figure 6.9 including the decomposed *A-Nodes*: *A1.1, A1.2, A2.1.1, A2.1.2*

6.2 Experiences Gained from the trial

We have in this section described the experience and knowledge gained through the application of our approach in the setting of our trial. Throughout the evaluation we have identified several weaknesses regarding the usability of our approach.

When applying our approach, we spent significant time in making the models of the system readable. The connections between both the components, categories, and the different nodes follows strict rulings, while the overall placement is not described other than the top-down approach. The structure of the system used in our trial allowed a simple separation between the different sub-trees (Figures 6.2, 6.3, 6.4) described in Section 6.1.2. It is unlikely the models would be as readable in a more complex system with significantly more overlap between the sub-trees.

The approach requires the *Security Analyst* to make several important decisions (E.g. categorization and groupings) early in the process that have significant influence on the outcome of the approach. This influence is not presented in the early steps, and a researcher without knowledge of the whole process might make assumptions requiring the revisit of the early steps later in the examination, costing both time and resources.

Several of the steps of the approach are limited in options, requiring the *Security Analyst* to follow strict requirements (E.g. component is lowest abstraction level). These requirements are applicable on the system used in our trial but are likely to affect the examination of systems with either a different structure or more complex components.

Throughout our application of the approach, we were required to revisit the research paper[8] describing the system on multiple occasions. This was due to either misunderstanding the role of a component, or to clarify its purpose within the system. A proper understanding of the components and their relationship ahead of the examination could prevent unnecessary delays in the early steps of the approach.

We used different bodies of knowledge in several steps of our approach and our ability to utilize these bodies significantly improved throughout the development of our approach. Existing knowledge of both different types bodies of knowledge, and their structure before applying our approach to a system is likely to affect both the quality of the results and time spent on the approach.

Chapter 7

Discussion

In Chapter 3 we described two different branches of research, along with several different evaluation strategies based on:

- Generalization
- Precision
- Realism

In this chapter we present threats to validity and reliability of our results and we have evaluated the development of the artefact described in Chapter 5 and how we used a *Trial of the approach* in the setting of a Smart Grid with self-healing properties as described in Section 6.1. We present how we used this trial to expand the approach and how it was utilized to improve the general process and modelling approach described in Chapter 5 through several iterations, basing the designed approach around our trial.

7.1 Threats to Validity

The limitations on time and resources, in addition to the compromises made with regards to our evaluation method introduces several threats against the validity of our results. In this section we discuss these validity concerns of our results and possible threats to both the validity and reliability of our approach and its result.

The lack of experience of the creator of the approach outside of an academic setting is likely to have had a significant impact on the usability of our approach. In Section 4.3.2 we described the inherent bias that comes with the use of a single investigator based on their background and prior experiences. The lack of practical experience outside of an academic setting is likely to have influenced our results and can harm the reliability of our approach in a natural setting.

A significant threat is introduced based on our involvement in both development and evaluation of our approach. This lack of separation between the developer and tester of our approach makes us unable to properly assess whether the approach, in its current form, is usable to a researcher without prior knowledge from its development. Until an independent researcher is successful in applying our approach in the setting of a smart grid system, we are unable to assess where the information presented in our approach sufficiently describes its use.

A threat to the realism of our approach is introduced through the categorization and abstraction of the components belonging to a system (Step 1 of the approach, Section 5.1).The created categories are an effort to describe the relationship between different components based on the structure of the system along with the relevant assumptions. However, this abstraction has the ability to remove characteristics from the components, or through the grouping of components add new characteristics that might not exist in reality. If the reasoning behind the choices made by the *Security Analyst* is not properly described and validated in Step 3 of the approach (Section 5.3), it has the ability to invalidate significant segments of the results. Misunderstood or misleading information early in the process can negatively affect both the precision and relevance of the results. This weakness in the approach is largely dependent on the understanding the actors involved in the process have about the impact of their choices throughout the approach.

The *Security Analyst* utilizes different models throughout the approach. These models contain several restrictions and are designed around the system in our trial of the approach. This presents a weakness in both the models and their rules, as they might not be applicable to a

system with a different structure. Both the models and the approach have limitations with regards to the abstraction of the components of a system: 1. During the decomposition of a system the approach requires the categories on similar abstraction levels, to be similar in scope. 2. In the approach we have defined the lowest abstraction level to be for components. This can result in a disproportionate amount of resources spent on simple components, presenting components of different scope and importance, as equals.

Our trial of the approach is based on the general description of a smart grid system with self-healing properties and the system was described in an academic paper[8]). Our approach has undergone several iterations during its development and has been significantly influenced by this paper. This influence might have had an unwanted impact on the relationship between our artefact and the results of our trial, harming the precision of our approach. The result can have affected the artefact in such a way that the approach underwent changes to fit the outcome of the trial, rather than the outcome of the trial changing alongside the approach.

As our approach exclusively has been applied in the context of an artificial setting (Our trial of the approach in Chapter 6) and since it throughout the process has undergone several iterations and significant change, the *generality* of our approach is significantly harmed. This «sample size of one» has a significant impact on our approach, as the approach has been tailored to the system used in our trial. The approach has not been applied to systems of different structure within the domain, and it can thus be argued that our approach is not general and only fitting for the specific setting in our trial.

We require in Step 1 of our approach (Section 5.1) that the involved actors make a choice regarding the expected time frame for the complete process. With the several iterations the approach has undergone, the time frame presented in our trial of the approach is not representative of the actual time we have spent on our trial. This harms the feasibility of our approach as we have not conducted our most recent iteration, from start to finish, on a secondary system, thus lacking data about the definite time usage of our approach.

In our trial of the approach, the role of *Domain Expert* was held by a research paper and in our approach, we described several functions for the *Domain Expert*. These functions have thus not been tested, making it difficult to assess the usefulness of the separate functions as well as the entire role of *Domain Expert*. It can be argued that a *Domain Expert* should be present if the *Security Analyst* is unfamiliar with the requirements of a smart grid, but we have not assessed the usability of the role and whether

a person would make a significant difference on the impact the role has in our approach.

We have not compared (in the same setting) our approach to an existing approach. The lack of a comparative analysis creates uncertainty regarding the usefulness of our approach compared to existing solutions within the domain. Additionally, while our approach has been developed in the setting of a smart grid, that does not necessarily signify its usefulness within the domain compared to more generic approach based around general cyber security.

7.2 To what degree are the success criteria fulfilled?

In Chapter 3 we described *Technology Research* and the iterative process based around three distinct steps:

1. *Problem analysis*
2. *Innovation*
3. *Evaluation*

In Chapter 2 we described the *Problem Analysis* through a general problem statement and six *Success Criteria* based around the actors:

- *Security Analyst*
- *Domain Expert*

In Chapter 5 we described the artefact created during the *Innovation*, and in Chapter 6 we further developed and utilized the created artefact in our *trial of the approach*. We have in this section evaluated the created artefact based on the *Success Criteria* and to which extent they were fulfilled in the setting of our *trial of the approach*.

7.2.1 Success Criteria 1

> *The testing approach is customized with respect to the specific needs of the Smart Grid Domain.*

In Chapter 5 we presented our approach, which was developed with a focus on the vulnerabilities existing within a system, and the use of selected *«bodies of knowledge»* to better utilize existing knowledge in the

process of security testing a system. The selected *«bodies of knowledge»* include generic checklists (E.g. *«OWASP TOP 10»*[9]) and bodies that to a bigger degree focus on IoT and industrial systems (E.g. *«OWASP TOP 10 IoT»[22], «Good Practices for Security of Internet of Things in the context of Smart Manufacturing»[26]*). We developed the initial approach through several iterations by applying our approach in the setting of a smart grid system with self-healing properties (Chapter 6). We can argue that the use of our trial in the setting of a smart grid system in addition to the use of domain specific *«bodies of knowledge»* to a lesser degree signify the relevance of our approach within the domain of smart grid.

In Section 7.1 we discussed the issue of generality with regards to our approach. While the method has been developed in the setting of a smart grid system, the smart grid system used is unlikely to be representative for the entire domain. Additionally, our trial of the approach was conducted in an artificial setting, with several limitations in both time and resource usage, harming the *realism* of our evaluation.

In summary, while the development and evaluation of or approach was conducted in the setting of a smart grid system, and the approach utilized several *bodies of knowledge* designed for the specific domain, our evaluation was severely limited. We are unable to present any stronger arguments with regards to the fulfillment of SC1 without additional empirical evidence.

7.2.2 Success Criteria 2

The testing approach must be resource and cost efficient.

In Section 5.1 we described the need to decide on both the amount of time and personnel to be used for our approach. The approach is designed around two roles, the *Security Analyst* and the *Domain Expert*. Requiring at minimum one *Security Analyst* and one *Domain Expert*. During our trial of the approach, the role of *Domain Expert* was held by a research paper describing the properties of the system and the general structure. While more resource efficient, this did however create difficulties were lack of familiarity in the specific setting might have resulted in unnecessary resource expenditure. The approach was designed around the two actors, and we lack data to properly assess whether the use of several actors within these roles would have a positive or negative effect on the speed and efficiency of our approach. While several *Security Analyst* could benefit the result of the approach in accordance with triangulation described in 4.3, the sequential nature of our approach makes it unlikely that multiple analysts would have a significant effect on the overall speed.

It can be argued that the emergence of Industrial IoT shortens the divide between smart grid systems and conventional ICT systems, thus

enhancing the usability of conventional approaches to cyber security within the smart grid domain. As presented in Section 7.1 we have neither conducted a comparative analysis between our approach and existing methods, comparing both the time and resource expenditure, or otherwise compared our approach to existing methods. Additionally, while we present the expected time usage in our trial of the approach, this assumption was not representative of the actual time usage. The iterative development of our approach removes any validity of the initial time usage assumption, and we are unable to accurately describe the divide between the time spent developing our approach and the time spent on evaluating said approach.

To summarize, the lack of a comparative analysis and data on the exact time spent on the trial of our approach, and without any additional evidence, we lack the necessary information to properly assess the degree of fulfillment of SC2.

7.2.3 Success Criteria 3

> *The testing approach must be generic enough for it to be of use on selected segments of a Smart Grid.*

In Chapter 5 we presented our approach. It utilizes a selection of *«bodies of knowledge»* which are general in nature and to a lesser degree applicable to both smart grid and conventional ICT-systems, with some differences in the expected use cases. We developed our approach based on a research paper presenting a general description of system within the domain of smart grid.

We designed the initial version of this approach independently from our trial of the approach, it was however heavily inspired by the research paper[8] our trial was based on. We described how the paper had a strong influence on both the development of our approach and the following evaluation in Section 7.1. This influence significantly harms the generality of our approach.

In summary, while our initial approach was outlined prior to the trial, one can argue that the approach has only been directly influenced by a single smart grid system. The inclusion of both generic and smart grid specific *bodies of knowledge* count for some external influence. However, the SC requires the approach to be *«of use on selected segments of a Smart Grid»*, which we are unable to fulfill without the study of our approach in the setting of several different systems.

7.2.4 Success Criteria 4

> *The testing approach must be viable during piloting of a Smart Grid.*

Our approach was developed and applied in the setting of a system based on a general description described in a research paper[8]. We are able to apply the approach on a simplified system, as proven by our trial. Additionally, the approach does not contain any specific requirements regarding the state (E.g. development, piloting, reproduction) of the examined system.

In Section 7.1 we described a weakness in how our approach utilizes abstraction, requiring components at the lowest abstraction level. We can argue that this constitutes a weakness in a piloting project where specific functions of a component are to be tested independently of the other components, as our approach has a bigger focus on the interconnectedness of the system.

In summary, while the approach is strict in the modelling of the different components, both our trial and the lack of specific requirements regarding the state of the examined system should not prevent its use in the piloting stage of development. However, we have not assessed whether the stage has an impact on the approach. We can thus argue that as our approach has not been tested on several systems, or systems in different stages of development, we are unable to assess the degree of viability our approach has in the given stage.

7.2.5 Success Criteria 5

> *The testing approach must result in unambiguous and detailed tests.*

In Section 5.5 we described the iterative process of the testing stage in our approach (Figure 5.6 on page 39), including the design of tests and the use of external resource to acquire either additional tests, or recommendations. These tests are based on possible vulnerabilities relevant to the different components within the examined system discovered throughout our approach. In Section 5.6 we present two documents created to assist the *Security Analyst* with organizing the different tests.

Our approach requires the *Security Analyst* to gather tests and test recommendations from external sources, which if applicable, are largely dependent on the source with regards to details and applicability. Additionally, the approach presents the *Security Analyst* with information allowing the creation of either component specific tests, or tests that are applicable to a range of components. However, the quality of the created tests based on this information is largely dependent on the knowledge and experience of the involved *Security Analyst*, and we have not outlined clear requirements for the content of the tests outside the documentation requirements presented in Section 5.6. Additionally, due to the objective of our trial (feasibility study), and limitations present due to the artificial

setting of our trial, we have been unable to properly assess the quality of the created tests.

To summarize, we can argue that the level of information our approach result in is dependent on the knowledge of the *Security Analyst*, making it difficult to assess. However, the setting of our trial limits the possibility of assessing the quality of the developed tests. We are unable to properly assess this SC until our approach has been applied in both a different context, and with different researchers performing the examination.

7.2.6 Success Criteria 6

Use of the testing approach must result in useful information.

In Chapter 5 we described our approach, which results in:

- A set of models describing the system.
- A set of tests in the form of multiple test documents, including test results (Described in Section 5.6).
- A test plan (Described in Section 5.6).

In Section 5.1 the *Security Analyst* was required to make choices regarding the structure of the system and the relationship between the different components and categories. The usefulness of the information gained throughout our approach is largely dependent on the abilities of the *Security Analyst* and their choices at that stage. Additionally, the different models and test documents created throughout the approach provide the *Security Analyst* with information that at a later time may be used to revisit the examination, redoing the later steps of our approach and continuously testing a system throughout its lifecycle.

In step 4(Section 5.4) and step 5(Section 5.5) of the approach the *Security Analyst* utilize the different *bodies of knowledge* to both gather information and references about possible weaknesses, and gather recommendations for tests and mitigations. We have neither assessed the quality of this gathered Information, or due to limitations in our evaluation assessed whether it is applicable to the examined system. The artificial setting of our trial constrained us to present general information, rather than information relevant to the specific components that would exist in a real system. Additionally, in Section 7.1 we described the possibility of a flaw early in the approach negatively affecting both the relevance and precision of the result of our approach. As we have not compared the results of our trial to those of a different approach, we are not able to assess the *realism* of our results.

In summary, the result of our approach is largely dependent on the *Security Analyst*, and while the models allows the *Security Analyst* to redo

the later stages of our approach and doing further testing at a later time, we have not assessed the quality of either the models or the overall results. While our approach results in the models and connected documentation, the focus of our trial has not been their quality, and we are thus unable to properly assess the fulfillment of this SC without additional empirical evidence.

Chapter 8

Conclusions

There exist various bodies of knowledge relevant to the testing of a smart grid system, both resources specialized for smart grid and other critical infrastructure, and more generic software security resources. In this thesis we have identified different bodies of knowledge presented in the *State of the Art* (Chapter 4) and we have proposed an approach to facilitate the security testing of a smart grid system utilizing both the knowledge of a *Security Analyst*, and existing knowledge present in the various bodies of knowledge. Our thesis has resulted in the following contributions:

- An overview of the state of the art
- An approach to security testing in the setting of smart grids utilizing various bodies of knowledge.
- A feasibility study of our contribution in the form of trial of the approach in the setting of a smart grid with self-healing properties.

Our approach consists of a general process presented in five distinct steps, and a modelling approach consisting of a CORAS[33] diagram and four distinct models designed around the specific needs of a smart grid system. The approach, presented in Chapter 5, outlines: The modelling of a system. An assessment to discover possible vulnerabilities that might affect the system through the use of various *bodies of knowledge*. The creation of tests from information gained through the modelling of the system and the information gathered from external resources. The testing of vulnerabilities that might affect the system.

We conducted a trial of the approach in the setting of a smart grid system with self-healing properties to investigate the feasibility of our approach and assess the application of our approach in this setting. Based on the results of our trial, we can argue that our approach is feasible in the setting the system assessed in our trial. However, due to limitations introduced through the artificial setting of our trial, the lack of empirical evidence regarding the application of our approach on different systems

within the domain, and the strong influence the described system has had on the development of our approach, we are unable to definitely conclude whether our approach is feasible on *«selected segments of a smart grid»*, and within the domain of smart grid and critical infrastructure.

8.1 Future work

Throughout our thesis we have identified several candidates for future work, we have prioritized the different candidates based on their effect on the limitations of our approach:

1. The application of our approach in a natural setting. Our trial of the approach was conducted in an artificial setting. Further studies in a natural setting is required to better assess the realism of our approach.

2. The application of our approach in the setting of several different systems. Our approach has been significantly influenced by the system described in [8] through both the development of our initial artefact and its application in our trial. Additional empirical evidence from the application of our approach in different settings is required to properly assess the generality of our approach.

3. A comparative analysis between our approach and existing solutions within the domain of smart grid. Throughout this thesis, we have not compared our approach to existing solutions. A comparative analysis is required to assess how our approach compares to existing solutions. This analysis should include both the cost/benefit aspect of our approach in comparison to existing solutions, and the quality of the result produced by our approach.

4. Expanding the use of existing information presented by the various bodies of knowledge throughout the approach. We have described the use of several bodies of knowledge to assist the *Security Analyst* in both discovering possible vulnerabilities and creating tests relevant for a system. The inclusion of additional bodies of knowledge could enhance the quality of the results from our approach.

5. Implement different approaches to better prioritize the discovered vulnerabilities. The use of our approach results in a set of vulnerabilities the system is affected by and we describe the option of ordering the discovered vulnerabilities either based on the total occurrences of a specific vulnerability or based on the number of vulnerabilities each component is affected by. Different approaches to the ordering

of discovered vulnerabilities could enhance the effect of our approach and the quality of our results.

6. The creation of a tool to assist the *Security Analyst* in the creation of the different models described in our approach. We have described and created a set of different models assisting the involved actors throughout our approach. A tool designed to both create and assess these models with regards to their described rules could both assist and speed up the use of our approach.

Bibliography

[1] Dustin Volz. "U.S. government concludes cyber attack caused Ukraine power outage". In: *Reuters* (Feb. 2016). [Accessed 13.02.19]. URL: https://www.reuters.com/article/us-ukraine-cybersecurity/u-s-government-concludes-cyber-attack-caused-ukraine-power-outage-idUSKCN0VY30K.

[2] Jewkes Polityuk Vukmanovic. "Ukraine's power outage was a cyber attack: Ukrenergo". In: *Reuters* (Jan. 2017). [Accessed 13.02.19]. URL: https://www.reuters.com/article/us-ukraine-cyber-attack-energy-idUSKBN1521BA.

[3] *Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures*. Tech. rep. European Union Agency For Network and Information Security, Nov. 2017. DOI: 10.2824/03228.

[4] Ida Solheim and Ketil Stølen. *Technology Research Explained*. eng. Tech. rep. 2007. URL: http://hdl.handle.net/11250/2387932.

[5] Erik Gøsta Nilsson. "FLUIDE: A Framework for Developing Flexible User Interfaces for Emergency Responders". PhD thesis. 2017.

[6] Michael D. Myers. "Qualitative Research in Information System". In: *MISQ Discovery* (June 1997). URL: https://www.researchgate.net/publication/220260372_Qualitative_Research_in_Information_Systems.

[7] Joseph E McGrath. *Groups : interaction and performance*. Englewood Cliffs, N.J, 1984.

[8] Aida Omerovic. et al. "A Feasibility Study of a Method for Identification and Modelling of Cybersecurity Risks in the Context of Smart Power Grids". In: *Proceedings of the 4th International Conference on Complexity, Future Information Systems and Risk - Volume 1: COMPLEXIS,* INSTICC. SciTePress, 2019, pp. 39–51. ISBN: 978-989-758-366-7. DOI: 10.5220/0007697800390051.

[9] *Owasp Top 10 2017*. [Accessed 27.03.19]. OWASP. URL: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_(en).pdf.pdf.

[10] *Enisa*. [Accessed 27.03.19]. Enisa. URL: https://www.enisa.europa.eu/about-enisa.

[11] *SANS*. [Accessed 27.03.19]. SANS. URL: https://www.sans.org/about/.

[12] *CISControls*. [Accessed 25.03.19]. CIS. URL: https://www.cisecurity.org/controls/.

[13] *Nist Cybersecurity*. [Accessed 27.03.19]. NIST. URL: https://www.nist.gov/topics/cybersecurity.

[14] *Mitre's Common Vulnerability and Exposure*. [Accessed 20.03.19]. Mitre's CVE. URL: https://cve.mitre.org/about/index.html.

[15] *Mitre's Common Weakness Enumeration*. [Accessed 20.03.19]. Mitre's CWE. URL: https://cwe.mitre.org/about/index.html.

[16] *Mitre's Common Attack Patterns Enumeration and Classification*. [Accessed 20.03.19]. Mitre's CAPEC. URL: https://capec.mitre.org/about/index.html.

[17] *MITRE*. [Accessed 26.05.20]. The MITRE Corporation. URL: https://www.mitre.org.

[18] *Mitre's Common Weakness Risk Analysis Framework*. [Accessed 21.03.19]. Mitre. URL: https://cwe.mitre.org/cwraf/data/vignettes-energy.html.

[19] *Mitre's Top 25 Most Dangerous Software Errors project*. [Accessed 22.03.19]. Mitre's CWE. URL: https://cwe.mitre.org/top25/index.html.

[20] *CWE/SANS TOP 25 Most Dangerous Software Errors*. [Accessed 01.03.19]. SANS. URL: https://www.sans.org/top25-software-errors.

[21] *The Open Web Application Project*. [Accessed 20.03.19]. OWASP. URL: https://www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project.

[22] *Owasp Top 10 IoT*. [Accessed 27.03,19]. OWASP. URL: https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project.

[23] *Securing Web Application Technologies [SWAT] Checklist*. [Accessed 27.03.19]. SANS. URL: https://software-security.sans.org/resources/swat.

[24] *CIS Controls Measures and Metrics for Version 7*. [Accessed 25.03.19]. CIS. URL: https://www.cisecurity.org/wp-content/uploads/2018/03/CIS-Controls-Measures-and-Metrics-V7.pdf.

[25] *ENISA Good practices for IoT and Smart Infrastructures Tool*. [Accessed 27.03.19]. Enisa. URL: https://www.enisa.europa.eu/topics/iot-and-smart-infrastructures/iot/good-practices-for-iot-and-smart-infrastructures-tool.

[26] *Good Practices for Security of Internet of Things in the context of Smart Manufacturing*. Tech. rep. European Union Agency For Network and Information Security, Nov. 2018. DOI: 10.2824/851384.

[27] *NIST Engineering Laboratory: Smart Grid*. [Accessed 02.04.19]. NIST. URL: https://www.nist.gov/engineering-laboratory/smart-grid.

[28] *NIST Cybersecurity Framework*. [Accessed 26.03.19]. NIST. URL: https://www.nist.gov/cyberframework.

[29] Mass Soldal Lund. *Model-driven risk analysis : the CORAS approach*. eng. Berlin ; 2011.

[30] B. Scneier. *Attack Trees*. [Accessed 23.04.19]. URL: https://www.schneier.com/academic/archives/1999/12/attack_trees.html.

[31] Clifton A. Ericson II. "Fault Tree Analysis - A History from the Proceedings of The 17th International System Safety Conferance - 1999". In: (1999).

[32] Polack F. Crivatanakul T. Clark J.A. *Effective Security Requirements Analysis: HAZOP and Use Cases*. Springer, 2004. ISBN: 978-3-540-30144-8.

[33] *The CORAS Language*. [Accessed 16.04.19]. SINTEF. URL: http://coras.sourceforge.net/coras_language.html.

[34] Rex Black. *Foundations of software testing : ISTQB certification*. eng. 2012.

[35] Schneider Schieferdecker Grossman. "Model-Based Security Testing". In: (2012). pages 1-12.

[36] Michael Quinn Patton. "Enhancing the Quality and Credibility of Qualitative Analysis". In: *Health Services Research* 34.5 pt 2 (1999), p. 1189. ISSN: 0017-9124.

[37] Workgroup 3. *INDUSTRY 4.0 AND ICS SECTOR REPORT - Cyber security for the industry 4.0 and ICS sector*. Tech. rep. industry-40-and-ics-sector-report-032018. European Cyber Security Organisation, Mar. 2018. URL: https://www.ecs-org.eu/documents/uploads/industry-40-and-ics-sector-report-032018.pdf.

[38] Guttorm Sindre and Andreas Opdahl. “Eliciting security requirements with misuse cases”. eng. In: *Requirements Engineering* 10.1 (2005), pp. 34–44. ISSN: 0947-3602.

[39] *Recommended Security Controls for Federal Information System*. Tech. rep. NIST Special Publication 800-53 Revision 1. NIST, Dec. 2006. DOI: 10.6028/NIST.SP.800-53r1.

[40] ISO/IEC JTC 1/SC 27. *Information technology — Security techniques — Information security management systems — Requirements*. Tech. rep. ISO/IEC 27001-2013. ISO, Oct. 2013.

[41] IoT Working Group. *Identity and Access Management for the Internet of Things - Summary Guidance*. Tech. rep. Cloud Security Alliance, 2016. URL: https://downloads.cloudsecurityalliance.org/assets/research/internet-of-things/identity-and-access-management-for-the-iot.pdf.