UNIVERSITY
OF TURKU

Design of risk assessment methodology for IT/OT systems

Employment of online security catalogues in the risk assessment process

Cyber Security
Master's Degree Programme in Information and Communication Technology
Department of Computing, Faculty of Technology
Master of Science in Technology Thesis

Author:
Anisia Spyrolari

Supervisors:
Professor Jouni Isoaho (University of Turku)
Postdoctoral Researcher Ali Farooq (University of Turku)
Prof. Dr. Eng. Fabio Massacci (University of Trento)
R&D Project Leader Enrico Schiavone (ResilTech S.r.l)

June 2021

Master of Science in Technology Thesis
Department of Computing, Faculty of Technology
University of Turku

Subject: Cyber Security
Programme: Master’s Degree Programme in Information and Communication Technology
Author: Anisia Spyrolari
Title: Design of risk assessment methodology for IT/OT systems: Employment of online security catalogues in the risk assessment process
Number of pages: 62 pages
Date: June 2021

Abstract

The revolution brought about with the transition from Industry 1.0 to 4.0 has expanded the cyber threats from Information Technology (IT) to Operational Technology (OT) systems. However, unlike IT systems, identifying the relevant threats in OT is more complex as penetration testing applications highly restrict OT availability. The complexity is enhanced by the significant amount of information available in online security catalogues, like Common Weakness Enumeration, Common Vulnerabilities and Exposures and Common Attack Pattern Enumeration and Classification, and the incomplete organisation of their relationships. These issues hinder the identification of relevant threats during risk assessment of OT systems. In this thesis, a methodology is proposed to reduce the aforementioned complexities and improve relationships among online security catalogues to identify the cybersecurity risk of IT/OT systems. The weaknesses, vulnerabilities and attack patterns stored in the online catalogues are extracted and categorised by mapping their potential mitigations to their security requirements, which are introduced on security standards that the system should comply with, like the ISA/IEC 62443. The system's assets are connected to the potential threats through the security requirements, which, combined with the relationships established among the catalogues, offer the basis for graphical representation of the results by employing tree-shaped graphical models. The methodology is tested on the components of an Information and Communication Technology system, whose results verify the simplification of the threat identification process but highlight the need for an in-depth understanding of the system. Hence, the methodology offers a significant basis on which further work can be applied to standardise the risk assessment process of IT/OT systems.

Keywords: cybersecurity, risk assessment, IT/OT, CWE, CVE, CAPEC, security principles, security patterns, attack-defense trees

Table of contents

Table of acronyms

IT / OT	Information Technology / Operational Technology
CPS	Cyber-Physical Systems
IoT	Internet-of-Things
PrOTectME	Protecting Operational Technologies of Medium Enterprises from Cyber Risks
CWE	Common Weakness Enumeration
CVE	Common Vulnerabilities and Exposures
NVD	National Vulnerability Database
CPE	Common Platform Enumerations
CAPEC	Common Attack Pattern Enumeration and Classification
FR	Foundational Requirement
SR	Technical Control System Requirement
CVSS	Common Vulnerability Scoring System
GrSM	Graphical Security Models
CSV	Comma-Separated Values
HTML	HyperText Markup Language
XML	Extensible Markup Language
JSON	JavaScript Object Notation
HWT	Hierarchical Weakness Tree
APT	Attack Path Tree
WVT	Weakness-Vulnerability Tree
S&R	Security & Resilience
GUI	Graphical User Interface
MQTT	Message Queuing Telemetry Transport
HTTP	Hypertext Transfer Protocol
ORM	Object-Relational Mapping
REST	Representational State Transfer
DSO	Distribution System Operators
EGC	Event Generation & Correlation
SMEs	Small and Medium-sized Enterprises
RISA	Risk Assessment in Security Argumentation
CSMS	Cyber Security Management System
OWL	Web Ontology Language
DL	Description Logics
SPARQL	Simple Protocol and Resource Description Framework Query Language
OVM	Ontology for Vulnerability Management

ELT	Extract, Load, and Transform

Table of figures

Table of tables

1 Introduction

1.1 Problem statement

The progress seen in Industry from 1.0, describing the mechanisation and steam power machinery, to 4.0, representing the implementation of cyber-physical systems (CPS) and Internet-of-Things (IoT) devices, has revolutionised industrial processes. Nevertheless, these improvements have also given rise to new issues, such as the vulnerability to cyber threats that exploit the IT aspect of CPS and IoT to compromise the functionality of the systems and the integrity and confidentiality of the generated data [1]. Operational Technology (OT) contains hardware and software that identify or cause modifications by monitoring and managing physical equipment and operations [2]. CPS are defined as combinations of computational systems that offer a deep interconnection with the associated physical entities and data-related functions accessed through the internet [3]. Hence, although the adoption of CPS and IoT offers many advantages concerning performance and productivity, this move expands cybersecurity vulnerabilities and attacks on OT.

However, assessing systems on their susceptibility to cybersecurity threats is a complex process considering the number of known vulnerabilities and attacks. Relying solely on penetration testing techniques to identify them is not an option in OT, provided that availability is crucial for productivity. In the context of risk assessment, several attempts have been made to create a centralised and organised collection of the potential threats. This attempt has led to creating online security repositories available to the public that list and group the weaknesses, vulnerabilities, and attack patterns. Instances of such catalogues are the Common Weakness Enumeration (CWE) [4], Common Vulnerabilities and Exposures (CVE) [5], the National Vulnerability Database (NVD) [6], the Common Platform Enumerations (CPE) [7], and the Common Attack Pattern Enumeration and Classification (CAPEC) [8]. At the time of writing this thesis, these catalogues stored from 527 entries, as seen in CAPEC, to 153,838 entries, as seen in CVE, each containing numerous descriptive information that ranges from definitions to examples and mitigations. This amount of data and the general scope they cover make identifying the established relationships and mapping the relevant entries to a given system under evaluation challenging while limiting the process to manual tasks. Although the catalogues provide categories and the option of a word-based search, two main problems are still present. Firstly, using the search bar requires selecting the appropriate keywords for the system under evaluation to ensure that all the relevant entries are considered. Secondly, even if the search bar is in use, the user must manually check numerous weaknesses, vulnerabilities, and attack patterns to identify the relevant information, which is a very time-consuming task, so an added filtering method is required.

As a response to the rise of cybersecurity issues in OT, many companies and organisations have taken initiatives to mitigate the effects of these issues; such an example is the “Protecting Operational Technologies of Medium Enterprises from Cyber Risks” (PrOTectME) [9]. This project was set up aiming at the definition of the theory and development of methods to create a cyber risk estimation service for digitalised and IT/OT 4.0 companies, which contain direct and indirect assets that are affected by cyber risks. The end goal of ProTEctME is to provide an automatic risk assessment process for IT/OT systems of small and medium-sized enterprises (SMEs) and automatic financial estimation of the cascading effects that any cyber-related attack or incident can have on the enterprise. Apart from this atomisation, the project offers services that assist enterprises’ compliance to standards related to their respective fields. One of the tools created and employed in PrOTectME is ResilBlockly, a system-of-systems modelling tool [10].

1.2 Objectives

Considering the problems identified in section 1.1, the main objective of this thesis is to provide a methodology that aims at simplifying and standardising the use of the online catalogues presenting the weaknesses, vulnerabilities, and attack patterns in the risk assessment of IT/OT systems.

The design of the methodology is based on three sub-objectives, whose aims are:

- To identify all the relevant relationships from the information in the online security catalogues
- To present a mechanism through which users can directly connect information across different catalogues
- To simplify and refine the mapping process between the assets of a system and the information found in the catalogues

1.3 Proposed solution

To achieve the goals listed in section 1.2, a methodology is proposed, as depicted in Figure 1.1.

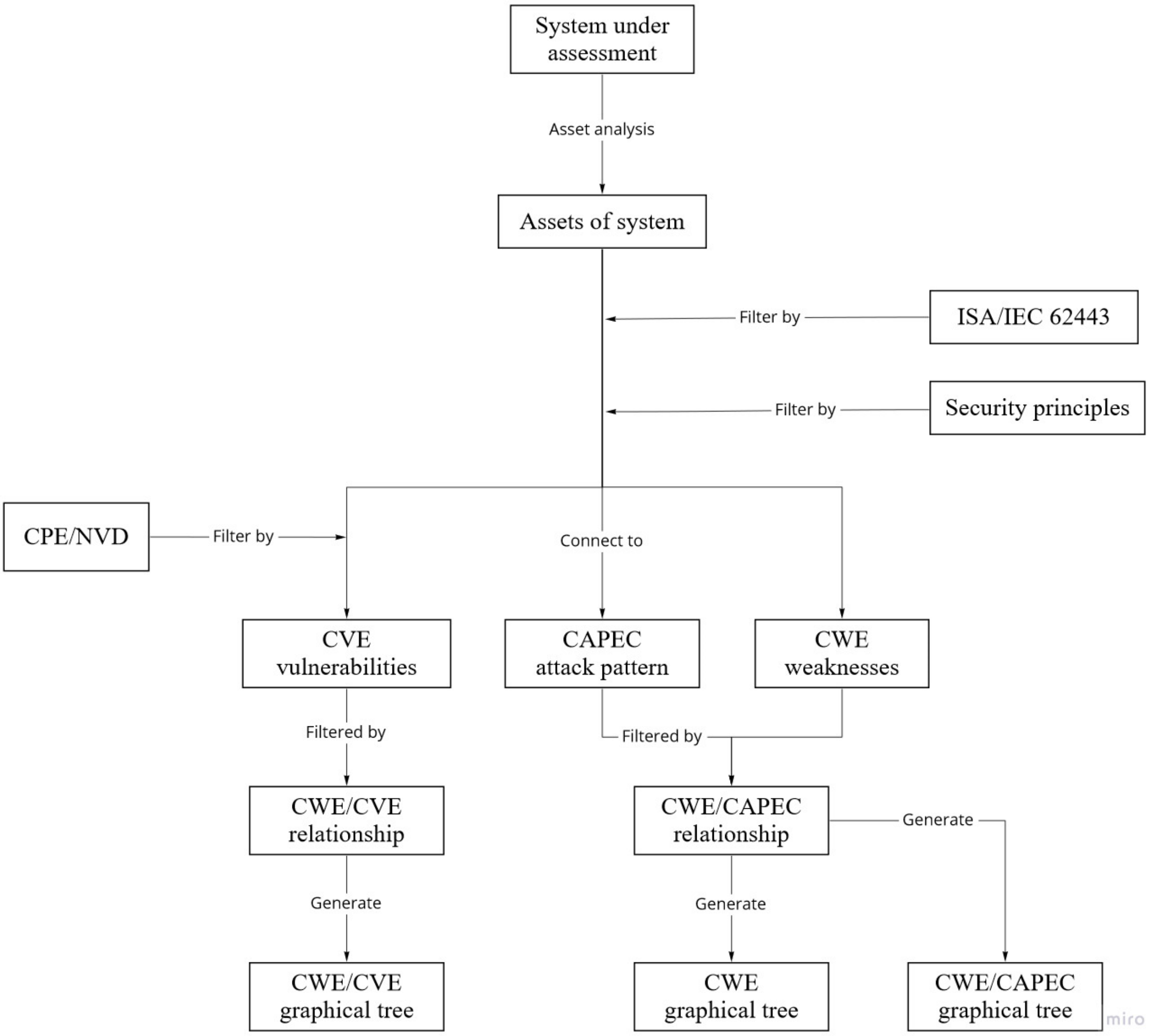


Figure 1.1: Overview of the methodology[1]

The methodology begins by hierarchically categorising the weaknesses and attack patterns found in CWE and CAPEC, respectively. The weaknesses employ the “ParentOf”-“ChildOf” relationship to provide the different levels of abstraction, while the attack patterns employ the “CanFollow”-“CanPrecede” relationship to present the path of attack patterns leading to an attack. After the online catalogues are organised, the assets are analysed and divided into components and subcomponents to simplify the user’s understanding of the security requirements with which each asset should be associated. The assets are initially characterised by the Foundational Requirements (FRs) found in ISA/IEC 62443-1-1 [11]. The assets under assessment are then connected to the CWE weaknesses and CAPEC attack patterns through the concept of security principles and the ISA/IEC 62443 standard. Security principles are defined as “distillations of experience designing, implementing, integrating, and upgrading systems that

[1] All the figures of this format are generated using MIRO. (http://www.miro.com/)

systems engineers and architects can use to guide design decisions and analysis" [12], particularly on issues of security determined by "speciality engineering disciplines", according to "Cyber Resiliency Design Principles" technical report generated by MITRE. Another definition can be extracted by the "Design Principles for Security" technical report generated by SecureCore, where security principles are presented as "guidelines or rules that when followed during system design will aid in making the system secure" [13]. This definition closely aligns with the Technical Control System Requirements (SRs) found in ISA/IEC 62443-3-3 [14], creating a link between security principles and the ISA/IEC 62443 standard.

Hence, the next step is mapping the CWE weaknesses and CAPEC attack patterns to the SRs, using their mitigations, and identifying subcomponents that group the mitigations in more specific categories. Once the weaknesses are connected to the asset, they act as links to the related attack patterns, and the same process is repeated for the attack patterns. Furthermore, the CWE weaknesses are connected to the security principles through their mitigations to connect them to security patterns. Security patterns are defined as "particular recurring security problems that arise in a specific security context and present well-proven generic schemes for security solutions" [15], according to M. Schumacher. Each weakness is connected to one or more security patterns through the security principles linked to their mitigations. The weaknesses are further connected to the related vulnerabilities found in CVE. As the number of available vulnerabilities is significant, CPE is used to filter the relevant ones through their link to NVD. NVD has a dual role; on one side, it provides a direct connection to the CWE catalogue, and on the other side, it offers the severity of vulnerabilities as provided by the CVSS. For each of the established connections, graphical representations are generated based on the concept of attack trees. Attack trees are tree-based Graphical Security Models (GrSM), which "graphically represent sets of attacks described in a hierarchical manner" [16][17]. The trees generated through the methodology present the hierarchical connection of weaknesses that lead to trees of sequential attack patterns. Similarly, the vulnerabilities can be presented based on their relationships to weaknesses. In order to test the methodology, the ICT Gateway use case provides its components as assessment targets to employ the methodology and evaluate its efficiency based on the results.

1.4 Thesis organisation

The rest of the thesis is organised into four categories, starting from the background information necessary to understand the designed methodology. The background information is explored in chapter 2, which contains the presentation of the target system ResilBlockly, the online security catalogues, the relevant literature and the ISA/IEC 62443 standard. Chapter 3 presents the designed methodology, which includes the organisation of the weaknesses and attack patterns of CWE and CAPEC through the ISA/IEC standard, the filtering of the CVE vulnerabilities through CPE, their graphical representation and the potential application of the methodology in ResilBlockly. Chapter 4 combines the testing of the methodology on components of the ICT Gateway use case, the analysis of the results and a discussion on the impact on present and potential future applications. Lastly, chapter 5 contains the conclusion, which summarises the thesis.

2 Background

This section provides insight into information used in the design and testing of the presented methodology. It introduces ResiBlockly, the target system of the methodology, the online security catalogues and their implementation in the ResilBlockly, and the relevant literature on which several concepts used in the methodology are based, such as academic papers and the ISA/IEC 62443 Standard.

2.1 Target system ResilBlockly

ResilBlockly is an updated version of the Blockly4SoS tool generated by the AMADEOS project [18]. ResilBlockly a tool that provides modelling, validation, query, and simulation functionalities for system-of-systems [19], focusing on cyber-physical systems. The tool offers two main features in the modelling phase: the “Profile Designer”, to design an abstract draft of the elements and their connection in a given environment, and the “Model Designer”, to specialise the “profile” to a particular event [10]. The profiles in the tool are created by utilising “building blocks” defined as “Class”, “Attribute”, “Relation”, “Menu”, and “Item Menu”, as seen Figure 2.1. This approach is based on the application of the “Blockly” library to the AMADEOS project [18].

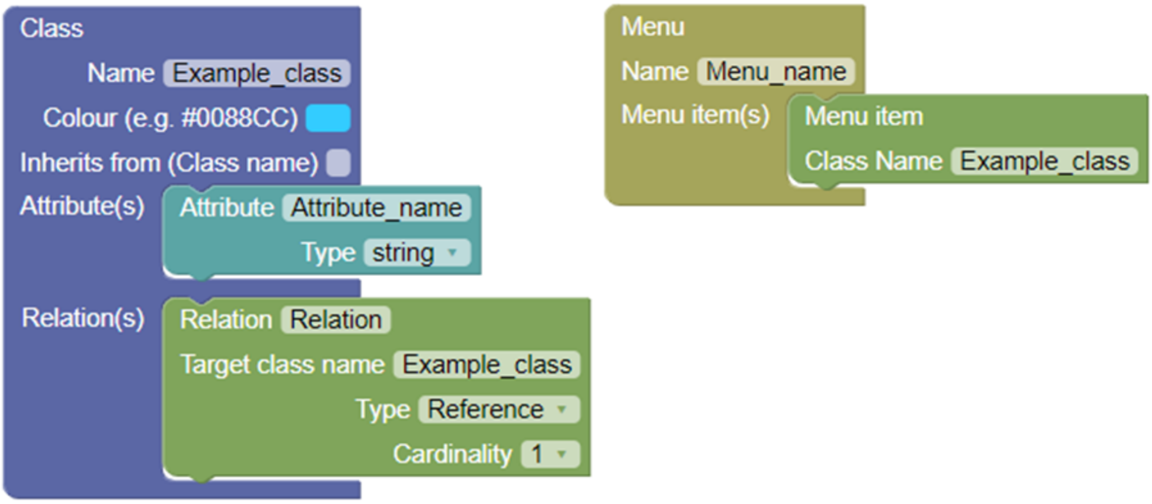


Figure 2.1: “Block” modelling elements used in ResilBlockly

An essential functionality of ResilBlockly is the use of the “Risk Designer” found in the “Profile Designer”. This process begins with the design of the “profile” followed by the selection of the weaknesses and vulnerabilities that might render a given “Class” vulnerable. The selection process of these threats is carried away with a "keyword" search over the information found in the CWE, CVE and CAPEC online security catalogues [10].

2.2 Online security catalogues

The methodology that will be introduced in chapter 3 employs the information found in five interconnected online security catalogues that manage security concepts like weaknesses, vulnerabilities, and attack patterns. The catalogues are community developed and offer material on the security of both software and hardware. These catalogues are the Common Weakness Enumeration (CWE), the Common Vulnerabilities and Exposures (CVE), the National Vulnerability Database (NVD), the Common Platform Enumerations (CPE), and the Common Attack Pattern Enumeration and Classification (CAPEC).

2.2.1 Common Weakness Enumeration (CWE)

The Common Weakness Enumeration (CWE) is a "community-developed list of common software and hardware" weaknesses [4] managed by MITRE [20]. Weaknesses are defined as "flaws, faults, bugs, or other errors in software or hardware implementation, code, design, or architecture that if left unaddressed could result in systems, networks, or hardware being vulnerable to attacks" [4]. For example, `"CWE-285: Improper Authorization"`, described as the lack of or inaccurately executed authorization control by a given software when a user seeks access to resources or permission to perform an operation [21], is a weakness found in the CWE catalogue. The approach of CWE is preventative as it aims to eliminate potential vulnerabilities at their origin by mitigating software and hardware mistakes before their exploitation. The information found in this catalogue can be utilized either by accessing the online platform or downloading it. CWE, depending on whether the entire catalogue or one of the predefined groupings is required, can be extracted in XML, CVS, or HTML format. The weaknesses listed in CWE are displayed alongside descriptions, points, and areas of introduction in the target's life cycle, consequences, mitigations, and detailed examples. Furthermore, connections are established to other related weaknesses, vulnerabilities, and attack patterns.

2.2.2 Common Vulnerabilities and Exposures (CVE)

The Common Vulnerabilities and Exposures (CVE) is a public catalogue of the up-to-date identified cybersecurity vulnerabilities [5]. Such an example is `"CVE-2021-28968"`, described as an XSS vulnerability identified in the “email” BBcode tag in PunBB before 1.4.6, which allows, under authentication, injection of arbitrary JavaScript into forum messages [22]. CVE is composed of listings of the vulnerabilities defined as CVE Records. CVE Records include three categories of information: CVE ID number, description of the vulnerability, and relevant references. The most notable piece of information found in CVE is the ID numbers, as they are utilized by “cybersecurity product and service vendors and researchers as a standard method for identifying vulnerabilities” [23]. The widespread use of these IDs directly links CVE to other online repositories, like the National Vulnerability Database (NVD) and the Common Platform Enumerations (CPE). The listed information of CVE can be reached using the online platform or by downloading the information in CSV, HTML, Text, or XML format.

2.2.3 National Vulnerability Database (NVD)

The National Vulnerability Database (NVD) is the “U.S. government repository of standards-based vulnerability management data represented using the Security Content Automation Protocol (SCAP)” [6] managed by the National Institute of Standards and Technology (NIST) [24]. In the NVD, each vulnerability has a description, an overview of its severity and the metrics that affect it, and further references to advisories, tools, and solutions. In addition, a connection is provided to related weaknesses found in CWE and affected software configurations found in CPE. The severity of a vulnerability in the NVD is determined using the Common Vulnerability Scoring System (CVSS) [25]. CVSS is “an open framework for communicating the characteristics and severity of software vulnerabilities” managed by FIRST [26]. The CVSS score in the NVD provides the base score linked to the vulnerabilities’ “innate characteristics”. Two versions of CVSS standards are used in NVD, v2.0 and v3.X. The severity in CVSS v2.0 [27] can be “low”, “medium”, or “high”, while in CVSS v3.X [28], it can also be “none” or “critical”. The exploitability metrics used to generate the base score are different in the two versions as well, but the impact metrics, consisting of “confidentiality”, “integrity”, and “availability”, remain the same.

Consequently, each vulnerability belongs to a severity category followed by a score between 0 and 10 and a vector string of the exploitability and impact metrics. The CVE ID numbers directly connect NVD and CVE, but both catalogues are necessary for a complete database of information on vulnerabilities. NVD may provide a vast number of details on the vulnerabilities, but it includes only a fraction of the vulnerabilities found in CVE. The information in NVD can be accessed through the online platform or downloaded in JSON format.

2.2.4 Common Platform Enumerations (CPE)

The Common Platform Enumerations (CPE) is a “structured naming scheme for information technology systems, software, and packages” [7] managed by NIST. CPE is derived from the “generic syntax” of the Uniform Resource Identifiers (URI). Hence, every entry consists of a “formal name format”, a method for checking names against a system, and a description pattern, which allows the association of text and tests to the name. More specifically, the CPE name is composed of numerous components that offer a more detailed description. Such components are the name, the vendor, the version, the update, and the edition. For example, “cpe:2.3:a:gnu:punbb:1.2.22:*:*:*:*:*:*:*” [29], as seen in Figure 2.2, is an entry of the CPE catalogue.

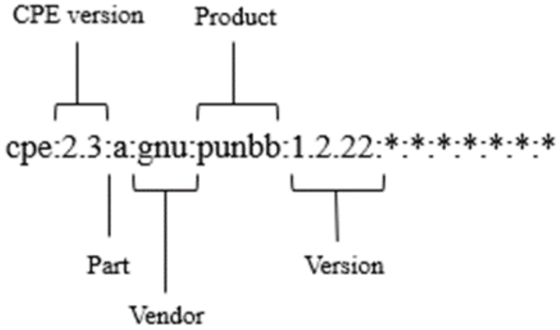


Figure 2.2: CPE entry

The public can access CPE as a dictionary, which provides online search or as an XML format.

2.2.5 Common Attack Pattern Enumeration and Classification (CAPEC)

The Common Attack Pattern Enumeration and Classification (CAPEC) is a public “dictionary of known attack patterns” [8] managed by MITRE. It is a guide through the exploitation techniques employed by adversaries against weaknesses and “cyber-enabled capabilities”. For example, “CAPEC-59: Session Credential Falsification through Prediction” is described as an attack that exploits anticipated session IDs, used in an activity to gain privileges and attempt spoofing or session hijacking attacks [30]. The attack patterns found in CAPEC provide an extensive description, an evaluation of the likelihood and severity of the attack, the required tools and skills to employ the attack successfully, the detailed attack steps to follow, their consequences and the potential mitigations. Apart from the individual overview, CAPEC provides lists of related attack patterns and related CWE weaknesses. Similarly to CWE, the information collected in CAPEC can be accessed through the online platform or by downloading them in XML, CSV or HTML format.

2.3 Relevant literature

The online security catalogues considered in this work are widely used for industrial applications and research purposes. As a result, many approaches are considered to categorize and connect the information found in them. These methodologies range from semantic applications of the CWE and CAPEC catalogues [31] to the implementation of “Natural Language Processing Techniques” [32] to directly link the attack patterns of CAPEC to the CVE vulnerabilities. The methodology designed in this thesis is based on the work and research presented in the following papers that combine the relationships drafted in the catalogues with the implementation of the concepts of security principles and security patterns.

2.3.1 Risk assessment

Risk assessment of IT/OT systems is broadly guided by security standards, whose use in the latter offers compliance to the identified security requirements and detection and mitigation of the threats and risks that impact them. Portela et al. [33] have implemented these goals in the Dutch DSO Enexis by presenting a Cyber Security Management System (CSMS) that combines the ISA/IEC 62443 and ISO/ISA 27001 standards. The process that was followed is presented in Figure 2.3.

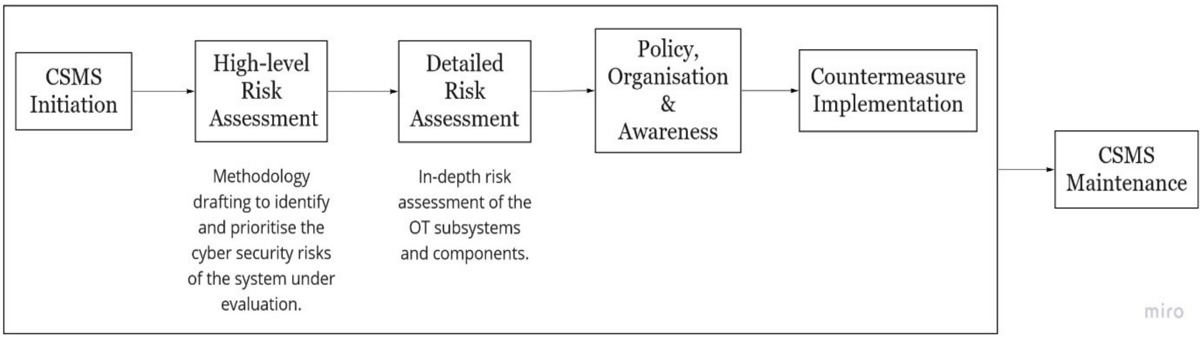

CSMS Initiation
High-level Risk Assessment
Methodology drafting to identify and prioritise the cyber security risks of the system under evaluation.
Detailed Risk Assessment
In-depth risk assessment of the OT subsystems and components.
Policy, Organisation & Awareness
Countermeasure Implementation
CSMS Maintenance
miro

Figure 2.3: Cyber Security Management System Overview [33]

Jelacic et al. [34] approach the potential need for Smart Grid OT Services to shift to a cloud-based environment by establishing a basis for assessing the risks that such a move may present. This method implements the ISA/IEC 62443 standard to divide the system into security zones and identify the threats that impact its confidentiality, integrity, and availability. The attack likelihood and severity levels are collected, creating a template for evaluating any Smart Grid system. An overview of this method is offered in Figure 2.4.

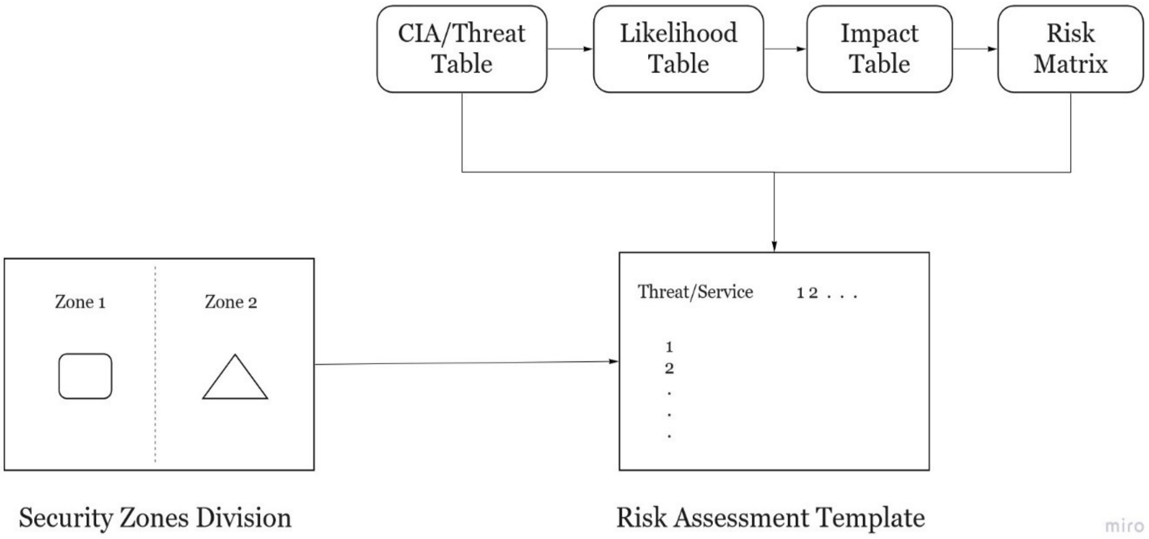


Figure 2.4: Smart Grid Risk Assessment (adapted from [34])

Based on the risk severity of the services, the decision is made to move the low and medium risk services and evaluate the structure for the high-level ones.

2.3.2 Semantic models

Considering the issues that arise in selecting the appropriate weaknesses and attack patterns from the CWE and CAPEC catalogues, A.Brazhuk [31] proposes a semantic approach to organise and categorise the two catalogues and simplify the management of the significant number of entries they provide to the public. The semantic models are based on the information found CWE and CAPEC that describe characteristics like the method of detection of a weakness, such as “Manual analysis”, to the required skill level that an attacker should have to apply an attack pattern successfully, which ranges from “Low” to “High”. The implementation of these semantic models is achieved by representing them as a Web Ontology Language (OWL) ontology, from which information is extracted via Description Logics (DL) or Simple Protocol and Resource Description Framework Query Language (SPARQL) queries. The semantic model of CWE and CAPEC can been seen in Figure 2.5.

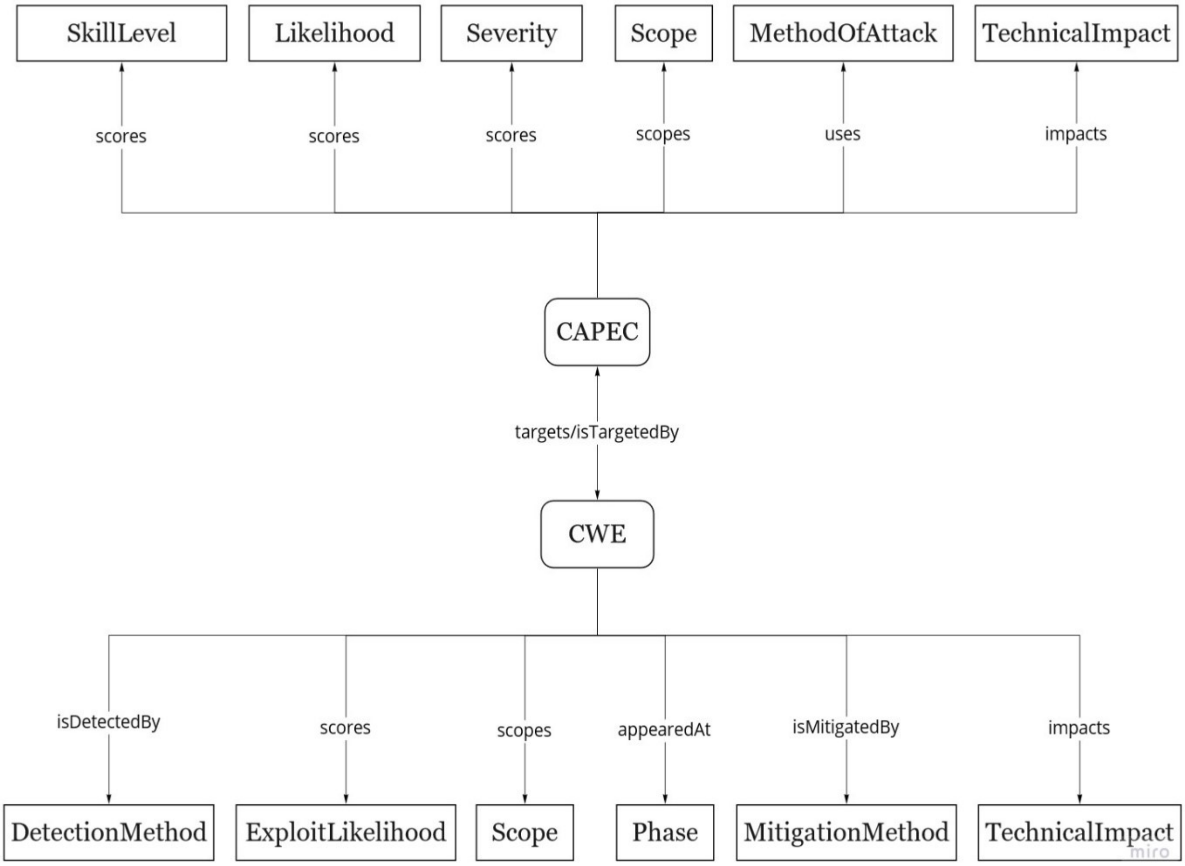


Figure 2.5: Semantic model of CWE and CAPEC [31]

2.3.3 Ontology of Vulnerability Management (OVM)

Wang et al. [35], motivated by the impact that vulnerabilities might have on the security of a system, drafted the Ontology for Vulnerability Management (OVM), which collects and applies attributes from security components of a system, like policies and countermeasures. This approach extracts data from online catalogues, like NVD, CVE, CPE, CWE and CAPEC, which map the potential threats and interactions. This mapping and an overview of the OVM is presented in Figure 2.6.

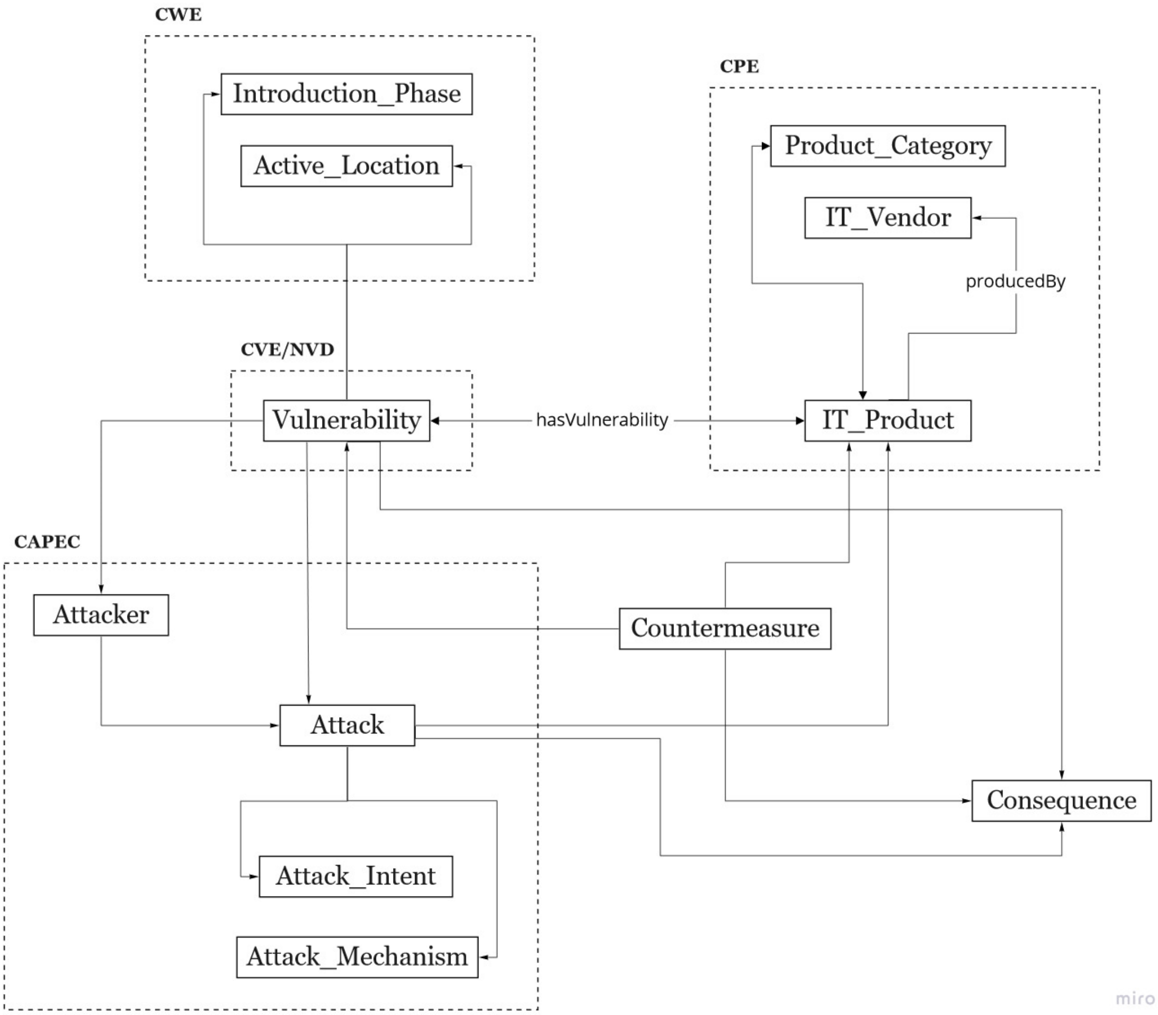


Figure 2.6: Ontology of Vulnerability Management Overview [35]

2.3.4 Security graphs

Online security catalogues like CWE, CVE and CAPEC manage information that belongs to different security concepts. However, the relationships between them are not readily evident to the users. To fill these gaps research is carried out by Xiao et al. [36], which provides a "security knowledge graph" that combines the weaknesses and attack patterns of CWE and CAPEC and the vulnerabilities of CVE. This approach, presented in Figure 2.7, serves a dual role, as it presents the links between the related security concepts and their instances and expands the security knowledge in predicting missing relationships in the entries of the security catalogues.

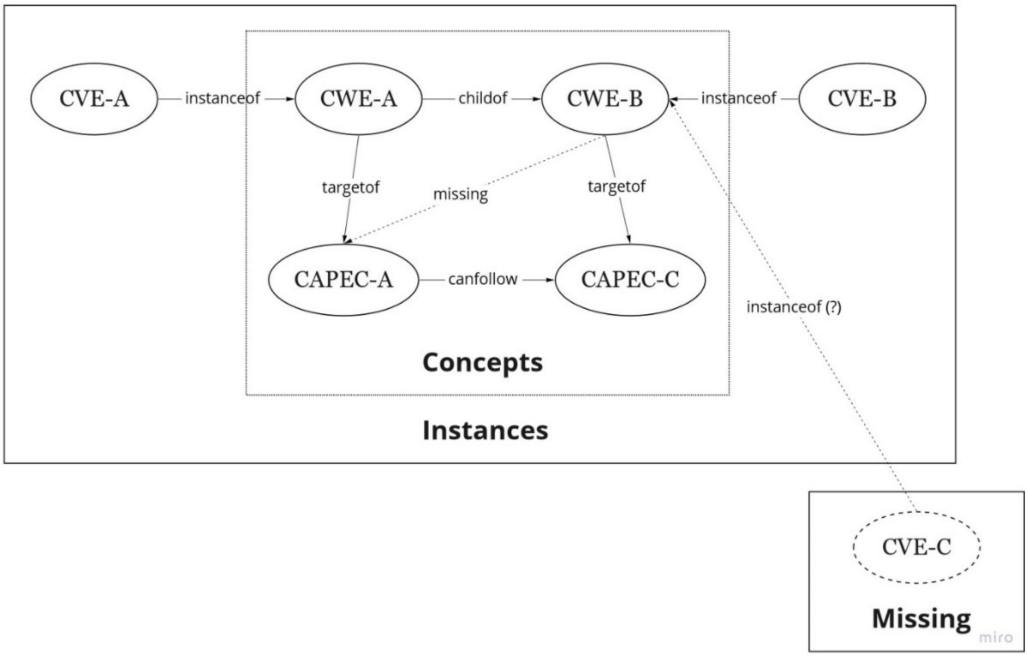


Figure 2.7: Security Knowledge Graph [36]

Another approach of graphically representing the connection between the information found in public security catalogues is seen in the research completed by Hemberg et al. [37]. This study employs the CWE, CVE, NVD, CAPEC, CPE catalogues and the “Tactics” and “Techniques” of the MITRE ATT&CK [38] catalogue. The information extracted varies in abstraction level, but it offers connections that might lead from the most general to the most specific concept. The existing links between the data in the different catalogues are manipulated by a graph framework named “BRON”, which graphically generates these layered connections, as seen in Figure 2.8.

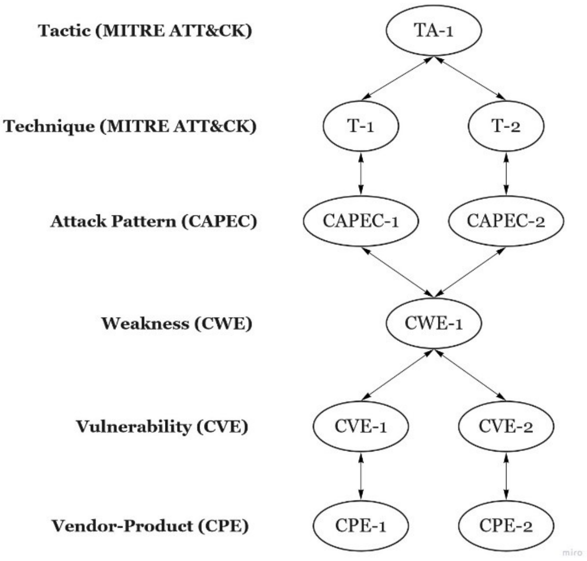


Figure 2.8: BRON Graph Overview [37]

2.3.5 Risk-based Security Argumentation (RSA)

The Risk Assessment in Security Argumentation (RISA) method, presented by Franqueira et al. [39], is employed to assist in the risk assessment process by applying the concept of argumentation in combination with the CWE, CVE, NVD and CAPEC security catalogues. RISA expands on the work provided by Haley et al. [40], which proposes connecting argumentation to risk assessment by presenting two types of arguments: the outer arguments that determine whether the operational environment of the system complies with its security requirements and the inner arguments which assess the validity of the outer arguments by challenging their basis.

The main goals of RISA are:

- to draft an approach that allows users to filter their decision with regard to the consequences of security risks and maintain a stable security level on their systems, and
- to identify the risk of a system and expand the source of the risk to the arguments considered.

The RISA method can be visualised in eight steps, as seen in Figure 2.9.

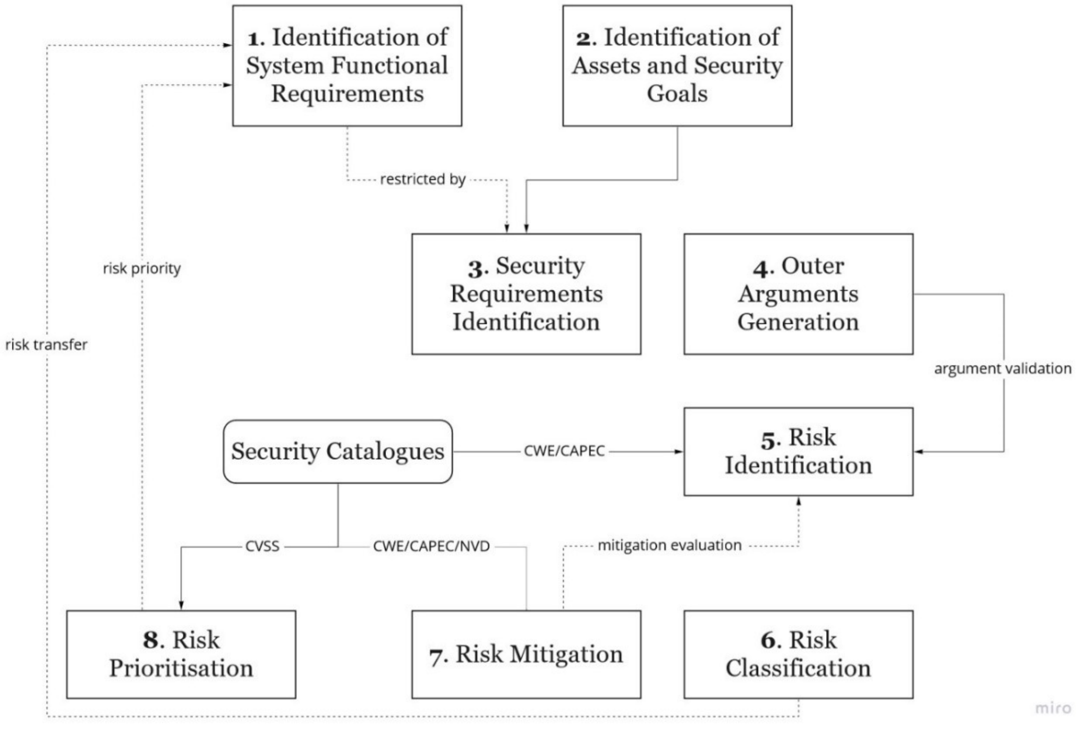


Figure 2.9: Overview of RSA Method [39]

2.3.6 Security patterns over CWE

Based on the work seen in “A classification methodology for security patterns to help fix software weaknesses” [41], a mapping is introduced between weaknesses found in the CWE catalogue to security principles and security patterns. The study leads to a semi-automatic methodology to classify security patterns found in the literature to simplify their selection and application. This methodology is composed of seven steps, starting with the hierarchical organization of the security principles derived from literature, followed by extracting the weaknesses and their mitigations from the CWE catalogue. Then, the security principles collected are mapped to the mitigations of the weaknesses; the security patterns are linked to the identified strong points; and through the latter, the security principles get related to security patterns. Hence, a database is constructed from which a given weakness offers the relevant security patterns. The steps of this method are presented in Figure 2.10.

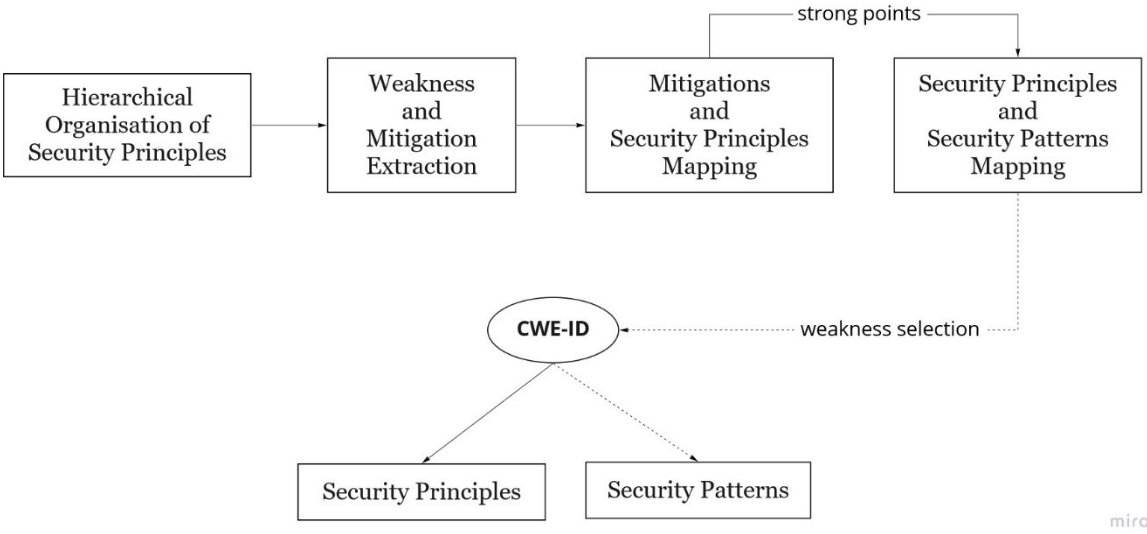

strong points
Hierarchical Organisation of Security Principles
Weakness and Mitigation Extraction
Mitigations and Security Principles Mapping
Security Principles and Security Patterns Mapping
CWE-ID
weakness selection
Security Principles
Security Patterns
miro

Figure 2.10: Security patterns over CWE Overview (adapted from [41])

2.3.7 Security patterns over CAPEC

The research presented in “A catalogue associating security patterns and attack steps to design secure applications” [42] is a continuation of the work presented in section 2.3.6, from the perspective of the attack patterns found in the CAPEC catalogue. The results obtained provide a semi-automatic methodology that leads to “Attack-Defense Trees” [43], graphically representing information on selected attacks and their potential mitigations. This methodology contains eight steps, starting from the extraction of information on attacks from the CAPEC catalogue, followed by hierarchical clustering of countermeasures collected from the CAPEC attack patterns, and a combination of the security patterns and the strong points selected. Then, the security principles are hierarchically organized and linked to the strong points and clustered mitigations. A database is created with the collected data that offers details on any chosen attack and the security patterns that offer potential solutions. This information can also be generated graphically, employing "Attack-Defense Trees". The steps of this method are presented in Figure 2.11.

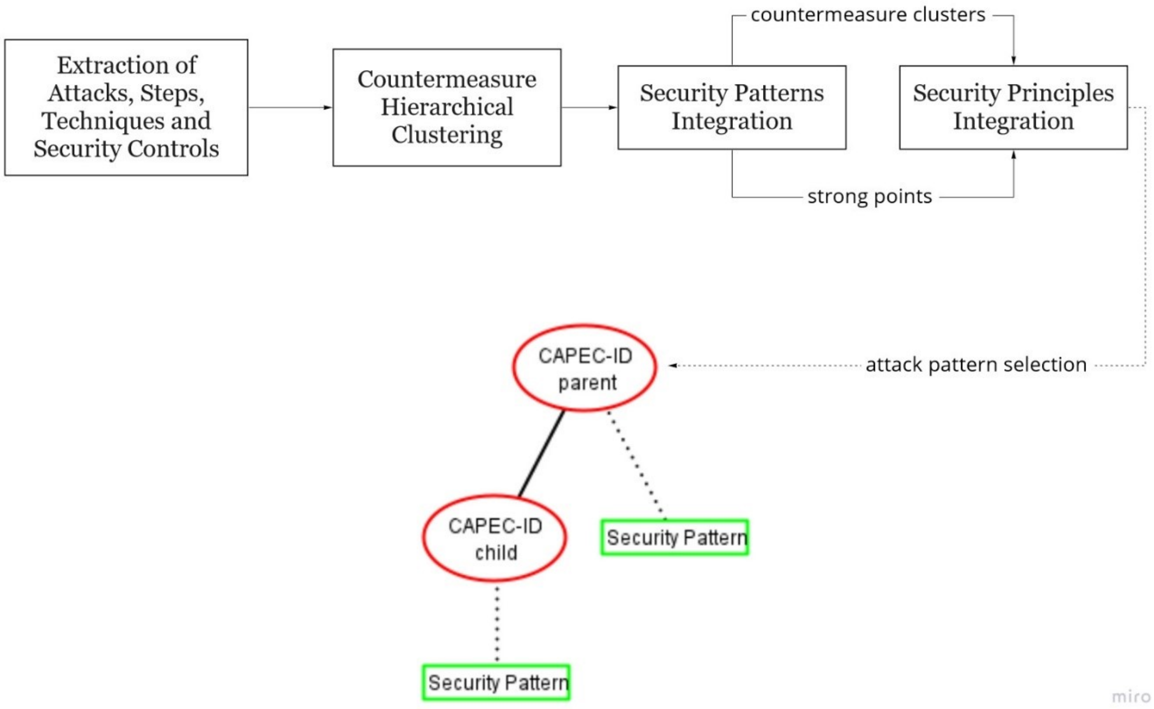

countermeasure clusters
Extraction of Attacks, Steps, Techniques and Security Controls
Countermeasure Hierarchical Clustering
Security Patterns Integration
Security Principles Integration
strong points
CAPEC-ID parent
attack pattern selection
CAPEC-ID child
Security Pattern
Security Pattern
miro

Figure 2.11: Security Patterns over CAPEC Overview (adapted from [42])

2.4 ISA/IEC 62443 Standard

The ISA/IEC 62443 standard is a standard drafted to manage the concept of cybersecurity in Industrial Automation and Control Systems. More specifically, the first part of the standard provides general insight into the target of the standard. There, the term "Industrial Automation and Control Systems (IACS)" is defined as the collection of "control systems used in manufacturing and processing plants and facilities, building environmental control systems, geographically dispersed operations such as utilities (i.e., electricity, gas, and water), pipelines and petroleum production and distribution facilities, and other industries and applications such as transportation networks, that use automated or remotely controlled or monitored assets [11]." The term "asset" in the ISA/IEC 62443 standard is used to define a system's resources for which protection is deemed essential. Therefore, a set requirement is the identification and listing of all the assets of a target system. The assets are grouped into physical, logical and human, which include the physical property of an organisation, the informational components associated with the operational functionality of the organisation, and the human element and their ability to perform critical tasks [11].

In order to describe security in IACS and the concepts that define it, some objectives are set. In information technology (IT), these goals are determined by the Confidentiality-Integrity-Availability triad (CIA). However, IACS are more complicated systems, making the CIA model inadequate. A notable difference in the objectives of security in IT to the one in IACS is their priority. In IT, security objectives have the hierarchical order: confidentiality, integrity and availability. However, in IACS, the main objective is to preserve the availability followed by the integrity, as the highest risks in this domain are associated with the control and management of the system's components. Confidentiality has the least priority, as the data analysed do not necessarily contain sensitive information. Hence, in the ISA/IEC 62443-1-1 [11] part of the standard, a more suitable alternative to the CIA model is presented for IACS. This model includes the following seven "Foundational Requirements":

1. The "Access Control (AC)" is the regulation of access to devices and information to preserve their examination only to authorised entities.
2. The "Use Control (UC)" is the regulation of the use of devices and information to preserve their activities only to authorised entities.
3. The "Data integrity (DI)" is the assurance of integrity for information found on communication channels to avoid unapproved modifications.
4. The "Data Confidentiality (DC)" is the assurance of confidentiality of information found on communication channels to avoid unauthorised intrusion or monitoring.
5. The "Restrict Data Flow (RDF)" is the regulation of the flow of information found in communication channels to avoid disclosure of sensitive material to unauthorised entities.
6. The "Timely Response to Event (TRE)" is the ability to alert the responsible authority of violations identified by relaying notifications containing the required forensic evidence. This process should initiate an automatic mitigating response that repairs "mission critical or critical safety situations."

7. The “Resource Availability (RA)” is the assurance of availability of all network resources to shield them from denial-of-service attacks.

An overview of the ISA/IEC 62443 IACS model is visualised in Figure 2.12.

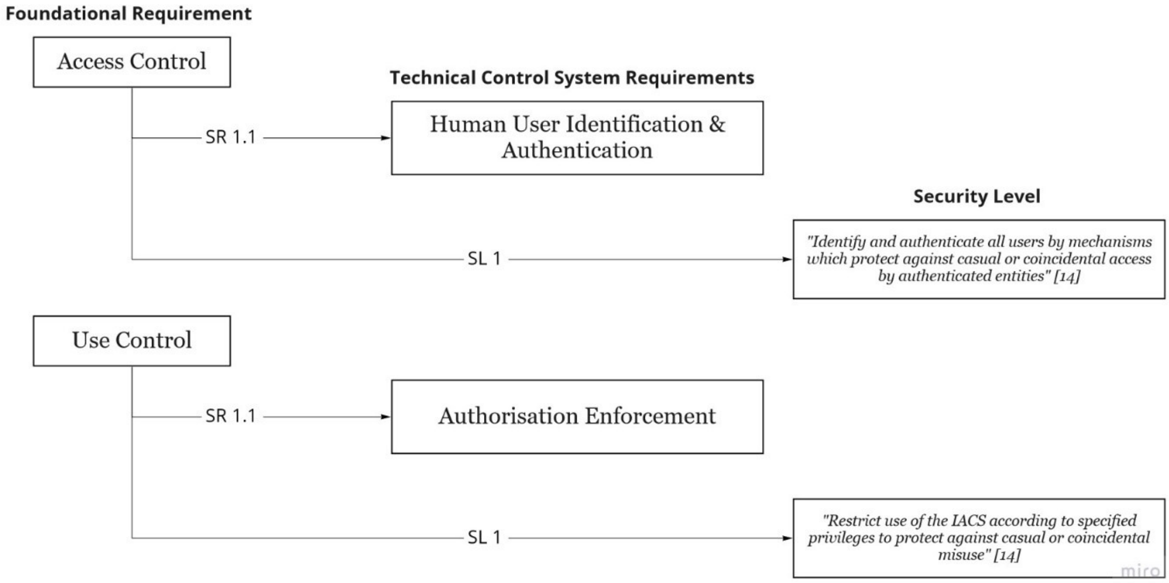

Foundational Requirement
Access Control
Technical Control System Requirements
SR 1.1
Human User Identification & Authentication
Security Level
SL 1
"Identify and authenticate all users by mechanisms which protect against casual or coincidental access by authenticated entities" [14]
Use Control
SR 1.1
Authorisation Enforcement
SL 1
"Restrict use of the IACS according to specified privileges to protect against casual or coincidental misuse" [14]
miro

Figure 2.12: ISA/IEC 62443 IACS model overview (adapted from [11])

As a system might have different sizes and levels of complexity, the security objectives will have to adapt to different levels of security. To achieve this categorisation, the ISA/IEC 62443 standard presents the concept of “zones”, which are defined as “logical groupings of physical, informational, and application assets sharing common security requirements [11].” The boundaries between the elements included and excluded from the zone are determined using borders. Another essential concept defined is the “communication conduit” as “a particular type of security zone that groups communications that can be logically organised into a grouping of information flows within and also external to a zone” [11]. Within conduits, communication is established through links called “channels” that share the equivalent conduit’s security properties [11]. So, in order to generalise the concept of security from individual devices or systems to zones, the concept of security levels is presented, with three types [11]:

1. The SL(Target) is the security level that a zone or conduit aims to achieve and is set during the risk assessment

2. The SL(Achieved) is the security level that the zone or conduit managed to attain

3. The SL(Capability) is the security level that the countermeasures affiliated to a zone or conduct are capable of or the ‘inherent’ security level that devices or systems of a zone or conduit can reach

Hence, a framework is presented that simplifies the decision-making process when countermeasures and devices with different security potential are concerned.

The ISA/IEC 62443-3-3 [14] part of the standard provides the security requirements for the targeted system. More specifically, after the zones and conduits are identified for a given control system and their respective SLs are determined, the Technical Control System Requirements (SRs) and their Requirement Enhancements (REs) form the additional requirements. This leads to a checklist for the system’s security requirements, where the seven foundational requirements categorise the SRs. A partial representation of this list can be presented using the Identification and Authentication Control, also referred to as Access Control. This FR has four SL-Cs that aim to identify and authenticate all the users that attempt to access the system and vary from protection against unintentional unauthorised access without technical skills to protection against targeted attacks with the required technical skillset. One of the SRs of this AC is “Human user identification and authentication” defined as the “enforcement of identification and authentication on all interfaces which provide human user access to the control system to support segregation of duties and least privilege under applicable security policies and procedures” [14]. This SR can be enhanced in the following ways: “unique identification and authentication”, “multi factor authentication for untrusted networks”, and “multi factor authentication for all networks” [14].

3 Designed methodology

This chapter provides a detailed description of the methodology designed to apply the information provided by the online catalogues to the risk assessment process of IT/OT systems. The segment is divided into seven parts that offer an overview of the methodology, a thorough explanation of the individual steps and the tools employed in the process, a proposed updating process and the application of the methodology in the target system.

3.1 Overview

The designed methodology is a path from selecting the system to assess to presenting the related weaknesses, attack patterns, and vulnerabilities. Figure 3.1 offers an overview of the methodology.

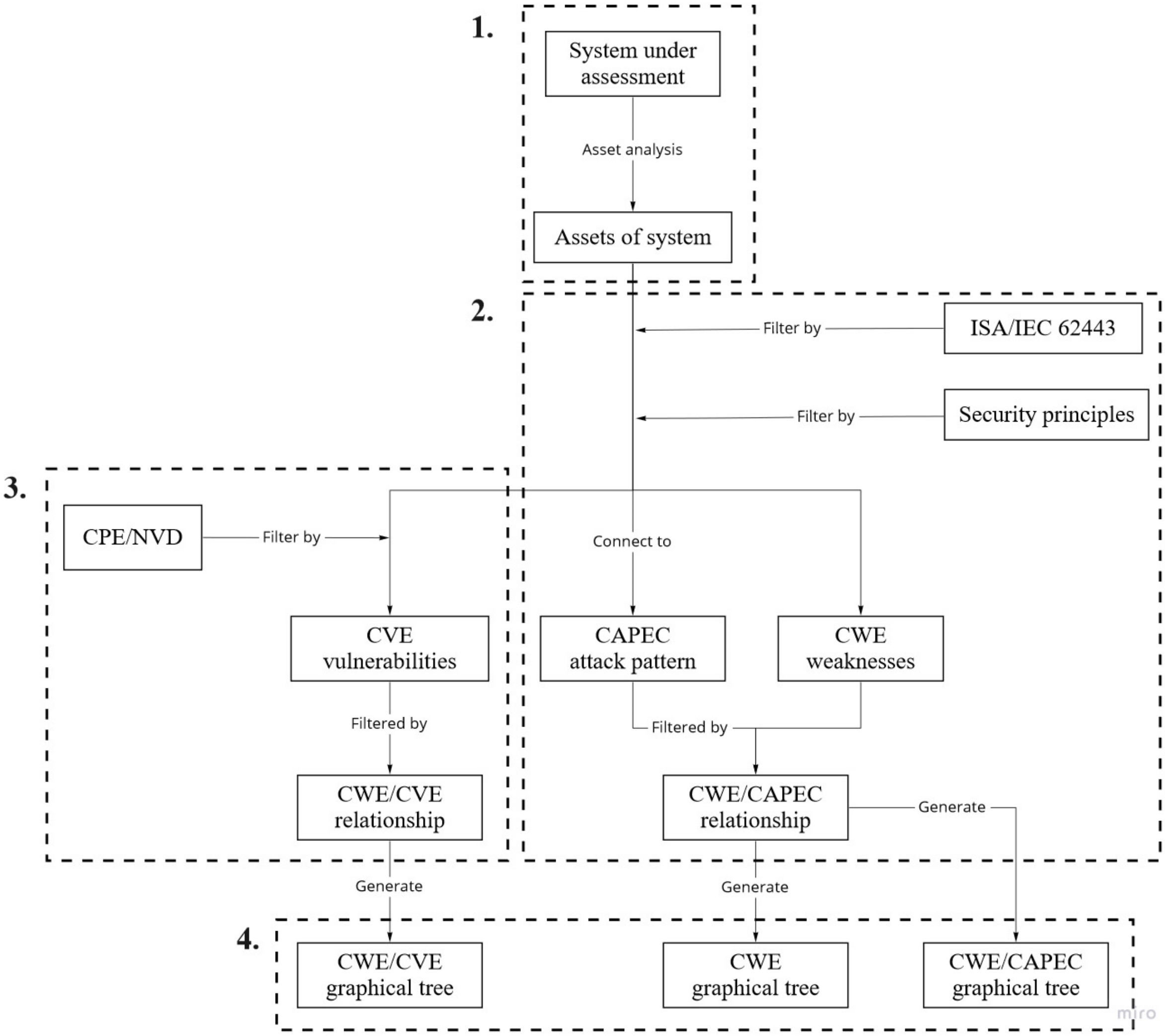


Figure 3.1: Overview of the methodology sections

The process can be described in four steps:

1. Division of the system in assets and components
2. Filtering of the CWE weaknesses and CAPEC attack patterns through the ISA/IEC 62443 standard and security principles
3. Filtering of the CVE vulnerabilities through the CPE and NVD
4. Graphical representation of the relationships established between the relevant weaknesses, attack patterns and vulnerabilities

3.2 Employed tools

In designing the proposed methodology, the Talend, ADTool and KH Coder tools were employed to extract, map and cluster information and graphically represent the results of the identified connections.

3.2.1 Talend Open Studio

Talend Open Studio [44] is an open-source "extract, load, and transform" (ELT) tool that assists in Data Integration and Big Data analysis. It is Eclipse-based and is used to produce and run ELT Jobs. [45] In the methodology design, Talend is used in every task that requires data extraction or connection based on a list of requirements and generation of outputs in different formats. More specifically,

1. Inputs are provided in XML, CSV, and Excel format
2. "tMap", a Talend component used to modify and lead data from one or more sources to numerous destinations [46], is employed to determine the required relationships between the inputs, or
3. "tXMLMap", a Talend component used to modify and lead XML data flow from one or more sources to numerous destinations [47], and
4. Outputs are generated containing the mapped data in XML and CSV format

In this process, the Talend Open Studio version 7.3 was used.

3.2.2 Attack-Defense Tree Tool (ADTool)

The Attack-Defence Tree Tool (ADTool) [48] is an open-source software created by the Security and Trust of Software Systems (SaToSS) group, part of the University of Luxemburg. ADTool is used to generate attack-defence trees that assist in graphical modelling and quantitative analysis. In the presented methodology, the concept of its use is expanded beyond

the attack-defence connection to comply with the identified relationships. It is critical to note that the changes are limited to the contextual use of the tool, as no technical alterations are implemented. The tool takes the XML files produced by Talend and generates a graphical representation of the relationships identified in the input.
In this process, the ADTool version 1.4 was used.

3.2.3 KH Coder

KH Coder is a publicly available software for "quantitative content analysis or text mining". [49] KH Coder is employed in the analysis process of the information found in the CWE and CAPEC catalogue to generate contextual clusters of the input data. The requirements set to the tool are the following:

- Stanford POS Tagger for the data "PRe-Processing",
- Ward Method with Jaccard Distance and TF-IDF for the "Cluster Analysis"

In this process, the KH Coder version 3 was used.

3.3 Asset connection to CWE

The starting point of the methodology is establishing a connection between the target's assets under evaluation and the weaknesses found in the CWE catalogue. This process is divided into five parts describing the sequential links leading from the identified assets to the relevant CWE weaknesses. These parts are: 1) the hierarchical organization of the CWE weaknesses, 2) the analysis of the system to identify the assets and their components, 3) the connection of the assets to the security requirements of the ISA/IEC 62443 standard, 4) the connection of standard to the weaknesses, and 5) the graphical representation of the results.

3.3.1 CWE hierarchical organisation

Based on the techniques and results examined in sections 2.3.6 and 2.3.7, the weaknesses found in CWE can be categorized based on the "Nature" of their relationship to other weaknesses. CWE defines eight such relationships: "ChildOf", "ParentOf", "MemberOf", "CanFollow", "CanPrecede", "Requires", "PeerOf" and "CanAlsoBe". This methodology employs three of them to generate connections between the weaknesses at different stages of abstraction. The "ChildOf", "ParentOf", and "MemberOf" can lead to a hierarchical representation of the weaknesses from the most to least abstractly defined.

In order to work with the information extracted from the CWE catalogue using the connections generated from their relations, a database is created storing all the data extracted:

- Using the XML version of the CWE catalogue as an input to Talend, the information related to the "ParentOf" - "ChildOf" are extracted.
- As the "ParentOf" - "ChildOf" relationship is sequential, the different categories can be grouped to form "levels" of relationships.

Hence, if a weakness is "ParentOf" but is not "ChildOf" of any weakness, the relationship would be part of "Level 0". However, if a weakness that is "ChildOf" in a "Level 0" relationship is also a "ParentOf" of another weakness, the relationship would be part of "Level 1", and the rest follow equivalently until there are no more "ParentOf" weaknesses. The database that stores these relationships is called "Hierarchical Weaknesses" database. An example of these relationships is provided in Figure 3.2.

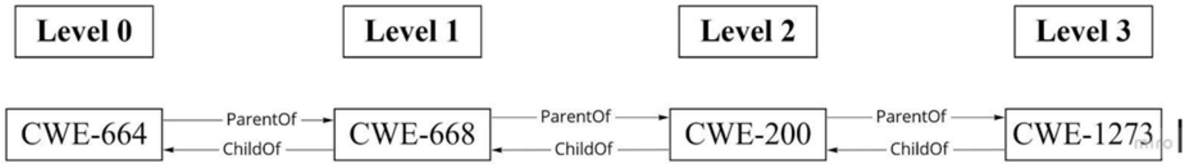


Figure 3.2: Example of "Hierarchical Weaknesses" database

3.3.2 Asset analysis

Assuming that the target of the risk assessment process is a system, the first step is to identify the assets that compose it. As assets can be considered systems as well, the next step would be to identify their components. Each component is described based on a list of technical details that provide insight into its composition, functionality, requirements, and restrictions. These technical details are used to create sub-components for each component, which are representative details that are repeated throughout the components of the asset. This effect is enforced considering that some components are different versions of the same core component, allowing their division into categories. The sub-components identified are essential in the risk assessment process as they can be used to extract the relevant weaknesses, attack patterns and vulnerabilities for the components they are part of by using some of the information found in the technical details as reference. More specifically, after collecting the technical details for all the system components, the similarities are identified, and a list is extracted containing the unique sub-components for each category. As some of the subcomponents serve the same functionality on every component they are found in, identifying the weaknesses, attack patterns and vulnerabilities associated with them is required only once. For the rest of the components, the results can be added automatically. If the functionality presents differences, then a base of common characteristics can be identified, limiting the number of weaknesses, attack patterns, and vulnerabilities to be checked for each component. Some categories of technical details, like libraries used for a given programming language, offer more detailed descriptions of the components and provide added sub-components or filters for the existing ones.

3.3.3 Asset to ISA/IEC 62443 standard

The assets of a system are connected to the ISA/IEC 62443 standard in two stages, as the entire asset and through the individual components. The mapping process starts by connecting the assets to the "Foundational Requirements" identified in the standard. As the FRs are general concepts, their relevance to the scenario under consideration is more evident. After the applicable FRs have been assigned to the assets, a more specialised mapping is enforced between the components of the asset and the "Technical Control System Requirements" derived from the selected FRs. Thus, for each component, the SRs relevant to the scenario under evaluation are assigned to them. This process is entirely manual, as it is based on the understanding of the assets, their components, and the security requirements that their functionality deems critical. The ISA/IEC 62443-3 lists seven FRs and fifty-one SRs divided among the FRs. So, the number of SRs that a user is required to consider depends on the relevant FRs.

3.3.4 ISA/IEC 62443 standard to CWE

Based on the definition of the security principles and the SRs, an assumption is made considering the SRs as security principles of the selected asset. However, security principles might contain one or more sub-principles that describe more detailed properties of the component. The security principles and the sub-principles act as links between the ISA/IEC 62443 standard and the CWE weaknesses. This process is based on the approach seen in sections 2.3.6 and 2.3.7, as it exploits the mitigations of the weaknesses. Hence, the process is the following:

1. Using the XML of the CWE catalogue as an input in Talend, the "descriptions" and the "strategies" of the mitigations of each weakness are extracted in relation to the CWE-IDs. So, for `"CWE-13: ASP.NET Misconfiguration: Password in Configuration File"` [50], the potential mitigations "Credentials stored in

configuration files should be encrypted. Use standard APIs and industry accepted algorithms to encrypt the credentials stored in configuration files.” is extracted.

2. The mitigations will be classified into sub-principles, creating a direct link between SRs and CWE weaknesses. Hence, CWE-13 is mapped to the “Use of Encryption” following the “Stored Data Encryption” and “Encrypted Credentials” sub-principles.
3. The methodology has three automatic levels of classification, starting by grouping the mitigations based on their IDs (MIT-ID).
4. The “Strategy” part of the mitigations is used to divide the entries further, based on the type of suggested countermeasure.
5. The rest of the mitigations are hierarchically clustered through text mining based on their degree of similarity, using the KH Coder tool on the “descriptions” of the mitigations.
6. As the classification is based on the context of the mitigations and the sub-principles, the final mapping demands manual contribution.
7. For the weaknesses that do not have mitigations in CWE, the mapping is based on their descriptions. Such an example is ‶`CWE-312: Cleartext Storage of Sensitive Information`″ [51], linked to the “Information Confidentiality” SR.

Following the clustering of the mitigations, each weakness in CWE is mapped to the relevant sub-principles and SRs. These connections, in turn, form a database that maps the SRs to the weaknesses through the sub-principles.

Table 3.1: Number of categories and weaknesses

	Technical Control	Security		
Foundational Requirements	**System Requirements**	**Sub-principles**		**Weaknesses**
1. Access Control	13	34	14	59
2. Use Control	12	17	28	165
3. Data integrity	9	90	51	538
4. Data Confidentiality	3	26	25	107
5. Restrict Data Flow	4	18	4	78
6. Timely Response to Event	2	27	20	111
7. Resource Availability	8	17	18	94

This mapping simplifies the process for the user, as the initial filtering of the relevant weaknesses takes place with the selection of the above categories. More specifically, by selecting the FRs, SRs and security sub-principles, the user limits the number of relevant weaknesses without analysing the individual weaknesses, as seen in Table 3.1. For that purpose, the connections between FRs, SRs and security sub-principles are mapped as seen in a partial representation in Figure 3.3. To be accessed and selected by the user, the CWE weaknesses are stored in the “CWE Security Principles” database mapped to the FRs, SRs and sub-principles they are related to.

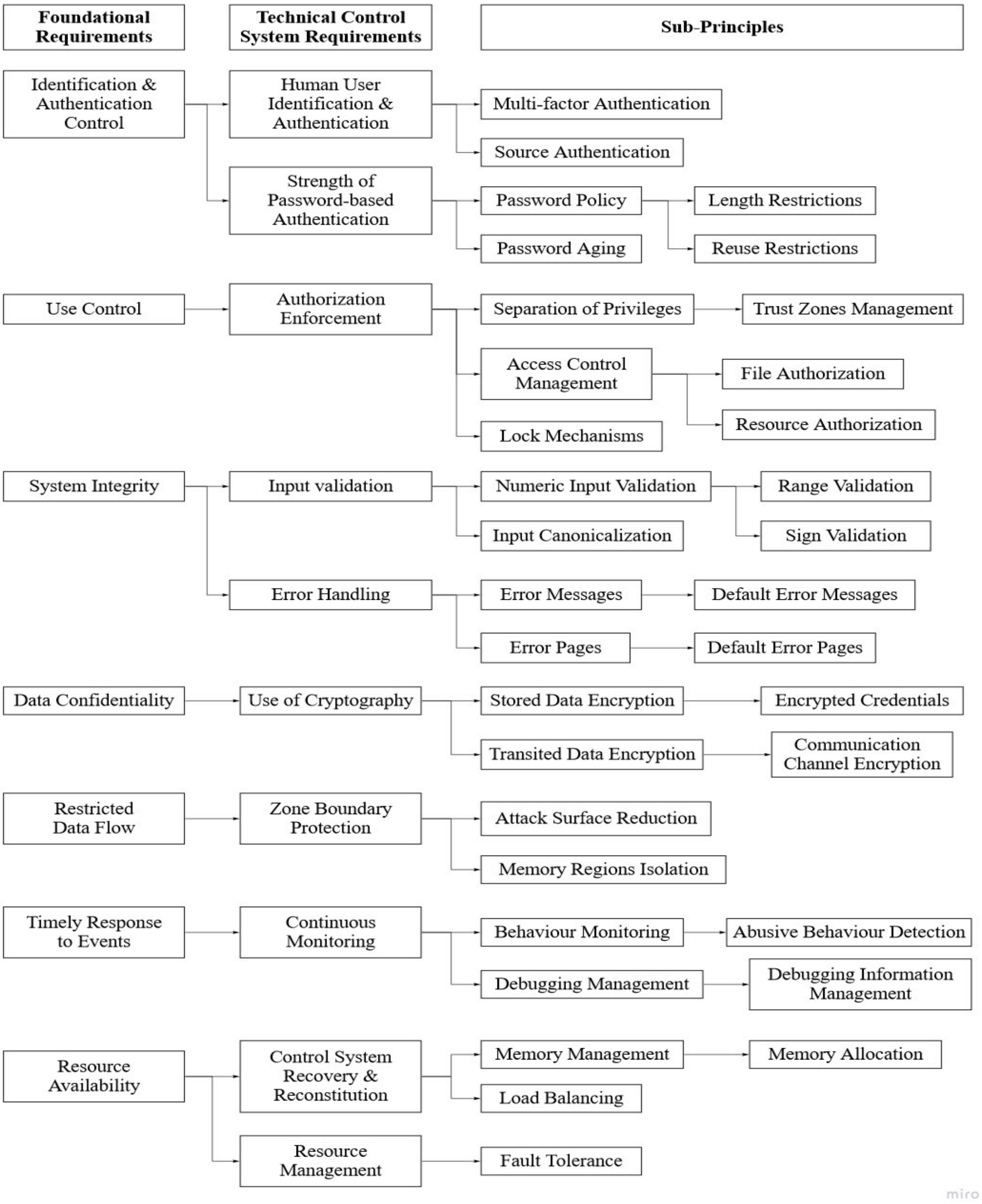


Figure 3.3: Categories of sub-principles derived from the ISA/IEC Standard

The filtering phase of the sub-principles is a combination of two steps as it allows the user to identify which of them are relevant to the asset while highlighting the ones already implemented in the asset as countermeasures. These steps remove the already mitigated weaknesses from the set of relevant weaknesses. Lastly, the user is asked to manually filter the identified weaknesses to check their compliance with the particular scenario by employing "keywords" to search for the required weaknesses. These "keywords" are extracted by the components, subcomponents and technical details describing the targeted asset.

3.3.5 CWE hierarchical trees

Based on the relationships identified in the catalogue study, four graphical tree versions were determined, one of which is the "Hierarchical Weakness Tree". The "Hierarchical Weakness Tree" is a type of tree-based only on weaknesses, categorised on the "ParentOf" – "ChildOf" relationship. In this case, considering the level of the chosen weakness, the tree can be expanded in one or two directions. If the weakness is the root, the tree provides nodes and leaves related to it, but if the weakness is a node or a leaf, the tree provides both the root and any other related weaknesses. More specifically,

- For each subcomponent, the identified weaknesses are mapped to the "ParentOf" weaknesses found in the "Hierarchical Weaknesses" database.
- For each "ParentOf" weakness, the related "ChildOf" weaknesses are identified and filtered based on the weaknesses identified as relevant to the component.
- A list of "ParentOf" – "ChildOf" weaknesses is extracted, where both categories are subsets of the identified weaknesses.
- When a complete and filtered list is generated, the relationships are mapped to their respective levels.

These relationships are identified and extracted using Talend.

Based on the outcomes of the mapping of the identified weaknesses to the hierarchical relationships established by the CWE catalogue, the "Hierarchical Weakness Trees" can be categorized on two criteria: the number of directions the tree has to expand towards and whether the targeted weakness has "ParentOf" – "ChildOf" relationships. These criteria present four types of "Hierarchical Weakness Trees":

- The "1-Direction HWTs", where the targeted weakness is provided with a tree that includes only "ChildOf" or "ParentOf" weaknesses related to it

- The “2-Directions HWTs”, where the targeted weakness is provided with a tree that includes both “ChildOf” and “ParentOf” weaknesses related to it
- The “Full HWTs”, where all the weaknesses in the “Hierarchical Weakness Tree” are relevant to the targeted component, and
- The “No parent HWTs”, where some weaknesses might not have a “ParentOf” weakness

Based on the relationships established, one or more types of “Hierarchical Weakness Trees” are generated for the targeted weakness found in the subcomponent using the XML file extracted by Talend and the ADTool to print the tree.

Figure 3.4 presents the incorrect management of access to resources of a software throughout its lifetime, from the most abstract "`CWE-664: Improper Control of a Resource Through its Lifetime`" [52] to the most detailed "`CWE-1273: Device Unlock Credential Sharing`" [53].

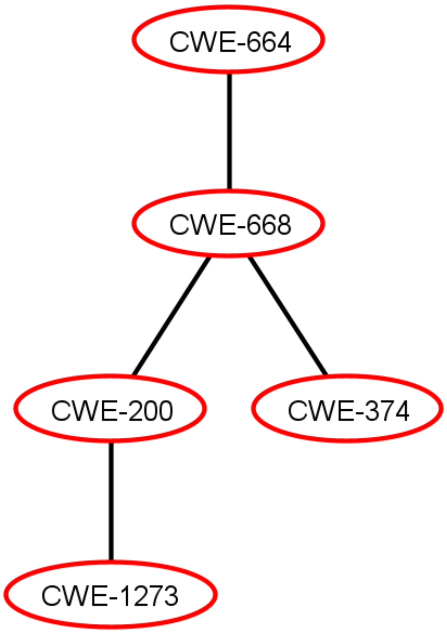


Figure 3.4: Example of partial “Hierarchical Weakness Tree”

3.4 Asset connection to CAPEC

Once the weaknesses have been organized and connected to the assets, a mapping is established linking the identified weaknesses and the attack patterns of the CAPEC catalogue that can exploit them. This process is divided into four parts describing two graphical tree models that illustrate the paths leading from an attack to the exploitable weaknesses and their proposed mitigations. These parts are the hierarchical organization and classification of the CAPEC attack patterns, the graphical representation of attack pattern chains, the link between CWE weaknesses and security patterns and the graphical representation including the weaknesses and security patterns to the chained attack paths.

3.4.1 CAPEC hierarchical organisation

Following the approach highlighted in section 3.3.1, the CAPEC attack patterns can be categorised based on the same relationships identified in the CWE weaknesses. The methodology employs two of the identified relationships to generate connections between the attack patterns at different stages of an attack. These relationships are the “CanFollow” and “CanPrecede”, which form attack pattern “chains” [54], where two or more attack patterns create a sequence in which one "can directly create the conditions that are necessary" to enable the exploitable properties of another attack pattern.

- Using the XML version of the CAPEC catalogue as an input to Talend, the information related to the “CanPrecede” – “CanFollow” is extracted.
- As the “CanPrecede” – “CanFollow” relationship is sequential as well, the different categories can be grouped to form the same types of relationship “levels” as the ones selected for the weaknesses.

Therefore, the respective “Level 0” of the attack patterns represents the relationship between attack patterns that do not follow any other attack pattern to the attack patterns that precede them. The following levels represent the same type of relationship between “CanPrecede” – “CanFollow” attack patterns. The process is documented in a database containing all the assembled attack pattern “chains” characterised by the level of their relationship. This database is called the “Attack Pattern chains” database. Furthermore, by using the relationship between CWE weaknesses and CAPEC attack patterns, the “CWE-CAPEC” database can be extracted from Talend, containing this relationship. Hence, paths can be formed from the attack pattern “chains” to the related weaknesses by combining the connections presented in the two databases, leading to the “Attack Paths” database. An example of these relationships is provided in Figure 3.5.

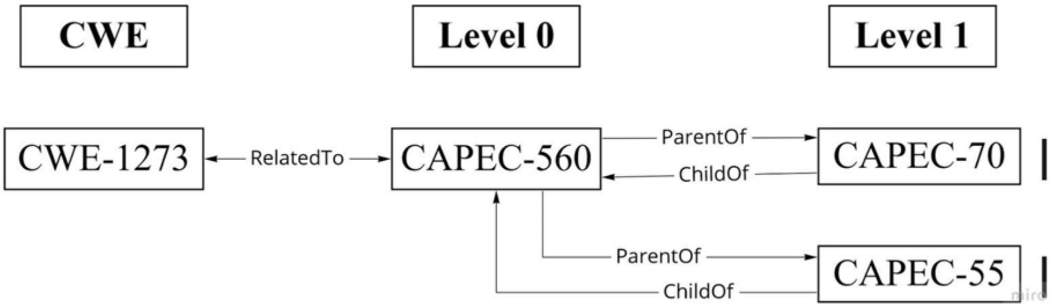


Figure 3.5: Example of “Attack Paths” database

3.4.2 ISA/IEC 62443 standard to CAPEC

Following the same process described in section 3.3.4, the attack patterns found in CAPEC can be classified based on the FRs and SRs of the ISA/IEC 62443 standard and the concept of security principles. More specifically,

1. Using the XML of the CAPEC catalogue as an input in Talend, the “descriptions” of the mitigations of each attack pattern are extracted in relation to the CAPEC-IDs. So, for “CAPEC-383: Harvesting Information via API Event Monitoring” [55], the potential mitigations “Leverage encryption techniques during information transactions so as to protect them from attack patterns of this kind.” is extracted.
2. The mitigations will be classified into sub-principles, creating a direct link between SRs and CAPEC attack patterns. Hence, CAPEC-383 is mapped to the “Use of Encryption”, defined as “Use cryptographic algorithms, key sizes and mechanisms for key establishment and management according to commonly accepted security industry practices and recommendations” [14] following the “Transited Data Encryption” sub-principles.
3. As CAPEC does not have “strategies” and mitigations IDs, the mitigations are hierarchically clustered through text mining based on their degree of similarity, using the KH Coder tool on the “descriptions” of the mitigations.
4. As the classification is based on the context of the mitigations and the sub-principles, the final mapping demands manual contribution.
5. The mapping is based on their descriptions for the attack patterns that do not have mitigations in CAPEC. Such an example is “CAPEC-629: Unauthorized Use of Device Resources,” [56] linked to the “Authorization Enforcement” SR.

Following the clustering of the mitigations, each attack pattern is mapped to the relevant sub-principles and SRs. These connections, in turn, form a database that maps the SRs to the weaknesses through the sub-principles. Hence, another database named “CAPEC Security Principles” is generated to store the CAPEC attack patterns mapped to the FRs, SRs and sub-principles they are related to.

This mapping has the same advantages for the user, as mentioned in section 3.3.4. Hence, the user filters the attack patterns by selecting the FRs, SRs and security sub-principles and manually selects the relevant ones based on their compliance with the particular scenario. The “keywords” employed for the search are the same ones extracted by the components, subcomponents, and technical details to identify the weaknesses.

3.4.3 CAPEC chain trees

Another version of the graphical tree identified is the “Attack Path Tree”. The “Attack Path Tree” is a type of tree that provides a listing of the potential paths that an attacker can follow to exploit a weakness by exploiting the established relationship between the CWE and CAPEC catalogues. The roots are the targeted weaknesses, and the nodes and leaves represent the related attack pattern “chains”, connected through the “CanPrecede” – “CanFollow” relationship and the CAPEC “Steps” that lead to each attack. The process of generating “Attack Path Trees” maps the assets, through their subcomponents, to the attack pattern that can exploit their weaknesses.

1. The attacks identified as relevant to the subcomponent are mapped to the initial attack patterns of the “chains”.
2. The extracted weaknesses are linked to the related attack pattern “chains” found in the “Attack Paths” database, and taking advantage of the hierarchical relationship of the weaknesses, a subset is extracted limited to the ones characterised as either only “ChildOf” or “ParentOf” without identified “children” weaknesses.
3. As not all attack patterns form “chains”, the identified ones are connected to the relevant weaknesses using the relationship marked in the “CWE-CAPEC” database.

Hence the generated “Attack Path Trees” provide the paths that reach the least abstract level of weaknesses in “ParentOf”-“ChildOf” relationships found in their hierarchical categorisation. Additionally, as there are attack patterns that are not connected to a weakness in CWE, these attack patterns are categorised as “Without identified weakness” results formed by Talend, as all the rest of the individual steps. Therefore, based on the relationships established, there are three versions of the "Attack Path Tree":

1. The "APTs based on attack pattern chains" provide a complete path from an attack pattern to the weaknesses targeted for exploitation.
2. The "Single attack pattern APTs" rely on direct connections between weaknesses and related attack patterns, and
3. The "No weakness APTs" focus on attack patterns that lack a connection to any weakness found in the CWE catalogue.

Based on the relationships established, one or more types of "Attack Path Trees" are generated for the targeted connection between weaknesses and attack patterns found in each subcomponent using the XML file extracted by Talend and the ADTool to print the tree.

Figure 3.6 presents that the unauthorised exposure of device unlocking credentials (`"CWE-1273: Device Unlock Credential Sharing"`) can be exploited by the `"CAPEC-560: Use of Known Domain Credentials"` [57] attack pattern, which is achieved by using `"CAPEC-55: Rainbow Table Password Cracking"` [58] or `"CAPEC-70: Try Common or Default Usernames and Passwords"` [59].

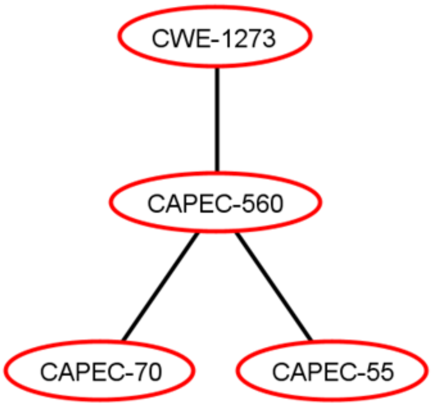


Figure 3.6: Example of partial "Attack Path Tree"

3.4.4 Security patterns mapping

Considering the number of weaknesses, attack patterns and their respective mitigations, security patterns provide a potential alternative to mitigating individual threats. Based on the results examined in sections 2.3.6 and 2.3.7, security principles provide a link between CWE weaknesses, CAPEC attack patterns and security patterns.

From publicly available online catalogues, [60][61][62][63] attack patterns were extracted and manually mapped to SRs or security principles or different abstraction, offering a link to CWE weaknesses. In this methodology, the attack patterns are connected to CWE, as according to CAPEC, the attack patterns are only viable if the related weaknesses exist in the targeted asset. For instance, "`CWE-653: Insufficient Compartmentalization`" [64] offers as the potential mitigation "Break up privileges between different modules, objects or entities. Minimize the interfaces between modules and require strong access control between them.". This mitigation is mapped to the "Authorization Enforcement" SR through the "Separation of Principles" sub-principle. Hence, a security principle connected to CWE-653 is "Privilege Separation", defined as the division of "one functional element into smaller functional elements with different privileges and restricted interfaces" [65].

Figure 3.7 offers a partial representation of the FRs, SRs and the related security patterns.

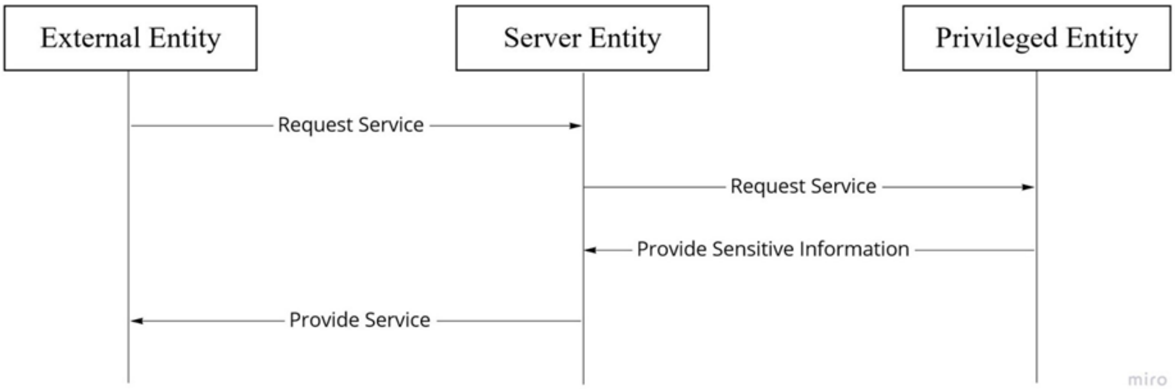


Figure 3.7: Example of "Privilege Separation" security pattern [65]

3.4.5 Attack-Defense trees

The last version of the graphical trees identified is the "Attack-Defense Tree". The "Attack-Defense Tree" is an expansion of the "Attack Path Tree", with the addition of the related weaknesses of the "Attack Path Tree" nodes and their security patterns. Hence, the process of generating "Attack-Defense Trees" maps the attack paths of an asset to the weaknesses that define them and their potential mitigations in the form of security patterns.

1. For each of the attacks contained in the attack paths, the related weaknesses are identified using the relationship marked in the "CWE-CAPEC" database, and
2. For each of the weaknesses, the related security patterns are identified.

Therefore, based on the relationships established, an “Attack-Defense Tree” is generated using the XML file extracted by Talend and the ADTool to print the tree.

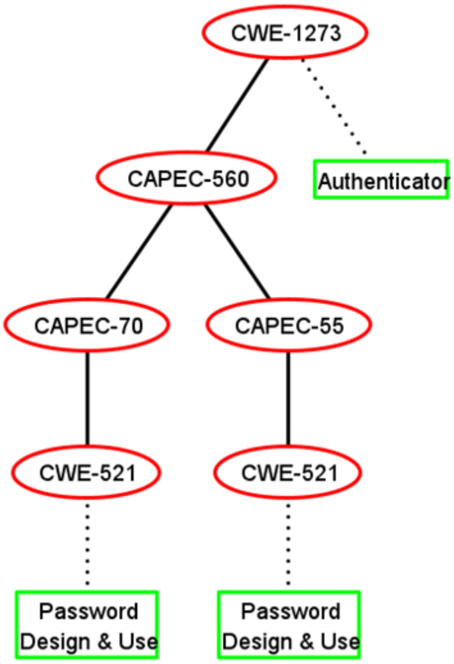


Figure 3.8: Example of partial “Attack-Defense Tree”

Figure 3.8 presents the security patterns that potentially mitigate “`CWE-1273: Device Unlock Credential Sharing`” and “`CWE-521: Weak Password Requirements`” [66]. The “Authenticator” security pattern is defined as “the problem of how to verify that a subject is who it says it is” [62], and the “Password Design and Use” security pattern is defined as “the best practice for designing, creating, managing, and using password components” [62].

3.5 Asset connection to CVE

Once the weaknesses and the attack patterns have been connected to the assets under evaluation, the related vulnerabilities in CVE and NVD catalogues are extracted. This process is achieved using the CPE catalogue to filter the vulnerabilities and graphically represent their relationship to CWE weaknesses.

3.5.1 CPE and NVD to CVE

Following the weaknesses and attack patterns, the CVE vulnerabilities are linked to the assets. The “keyword” search process is adopted utilising the exact keywords used for the weaknesses and attack patterns. However, the process is connected to CPE by adding information on the name, vendor, version and update of each subcomponent. More specifically,

1. Using the XML form of the CWE catalogue as input in Talend, extract the CVE vulnerabilities and the related CWE weaknesses.
2. Extract the vulnerabilities found in CVE and their “descriptions.”
3. Map the vulnerabilities of the CWE to the “descriptions” extracted from CVE.
4. Extract the vulnerabilities found in NVD and the related CPE entries and the related weaknesses.
5. Extract the name, vendor, version and updates of each entry in CPE.
6. Map the vulnerabilities to the information extracted by CPE
7. Connect all the extracted information and store them in a “CVE-NVD-CPE” database.

This process limits the number of vulnerabilities the user needs to filter manually and it can be expanded by extracting the severity levels and characteristics of the vulnerabilities as calculated by versions 2 and 3 of CVSS. Therefore, taking advantage of the connections identified, a link is determined between CVE weaknesses and the information provided by CVSS.

3.5.2 CWE/CVE/NVD relationship trees

Another version of the graphical trees identified is the “Weakness-Vulnerability Tree”. The “Weakness-Vulnerability Tree” is a type of tree that takes advantage of the relationship found in CWE between the weaknesses and the vulnerabilities found in CVE and NVD. In this tree, the root is the weakness, and the connected vulnerabilities are the leaves. The process of generating this tree is the following:

1. Extraction of the CWE weaknesses and the related CVE vulnerabilities linked through the CWE catalogue and their storage into the “CWE-CVE” database
2. Mapping the weaknesses identified for the subcomponents of the asset to the “CWE-CVE” database, the related CVE vulnerabilities are exported and can be linked to the CVE catalogue to extract the specific descriptions of the vulnerabilities
3. Reversion and repeat the extraction phase for the vulnerabilities generated from the CVE catalogue
4. Instead of using the weaknesses to identify the related vulnerabilities, the vulnerabilities are used to identify the weaknesses found in the same CWE relationships
5. As several vulnerabilities of CVE might be identified without a connection to the CWE weaknesses, the mapping process is repeated using the NVD catalogue, which provides connections between the vulnerabilities and the weaknesses in CWE.

Due to the generality of the weaknesses, the related vulnerabilities might cover broad concepts which are not necessarily relevant to the component they identified with, so the outcomes should be filtered based on the given scenario.

This process can be achieved semi-manually by using the CPE catalogue. Provided that the user has the name, vendor, version, update or all three for each subcomponent, the CPE can reduce the number of results of vulnerabilities and manually filter the rest.

Hence, based on the relationships established, there are two forms of the "Weakness-Vulnerability Tree":

- The "Full WVTs", where the vulnerabilities are connected to a relevant weakness through the CWE relationship or the NVD catalogue, and
- The "No weakness WVTs", where the vulnerabilities are not related to any weakness through the CWE relationship or the NVD catalogue.

Based on the relationships established, one or more types of "Weakness-Vulnerability Tree" are generated for the targeted connection between weaknesses and attack patterns found in the subcomponent using the XML file extracted by Talend and the ADTool to print the tree. A partial example of the "Weakness-Vulnerability Tree" is seen in Figure 3.9.

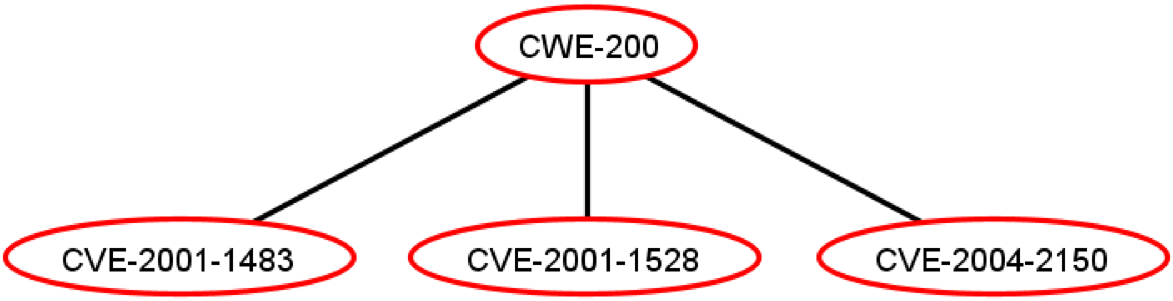


Figure 3.9: Example of partial "Weakness-Vulnerability Tree"

3.6 Updating process

The proposed methodology might offer the user a more manageable identification mechanism of the weaknesses, vulnerabilities, attack patterns, and relationships formed among them; however, the information it employs should be simply updatable for the methodology to be practical in the long term. Firstly, the updating process can be divided into two categories based on the part of the methodology that is being updated. As the databases employed in the methodology store information extracted from the online catalogues, their updates are linked to the updates of the online repositories. Hence, the updating process can be achieved in three steps:

1. Download the new versions of the security catalogues in the required format (XML, CSV, JSON, or HTML)
2. Input the downloaded versions of the catalogues in the Talend mappings created, and extract the updated databases
3. Replace the old databases with the new ones

Due to the automated form of these steps, the updating process is completed in minutes.

The second category focuses on updating the databases containing the classification of weaknesses and attack patterns based on the security principles, and security patterns are equally simple, but they are semi-automated. The automated part follows the algorithm seen in Figure 3.10, where based on the "Security Principles" databases of the current version of CWE or CAPEC, the new versions of the catalogues are compared and linked to the relevant groups.

```
Algorithm: Update
Upload DB_v.4.3, DB_v.4.4
Declare new_DB_v.4.4
For x in DB_v.4.4:
    For y in DB_v.4.3:
        If x = y:
            Add x to new_DB_v.4.4
            Add the security principles of y to new_DB_v.4.4 for x
            Remove x from DB_v.4.4
Download new_DB_v.4.4, DB_v.4.4
```

Figure 3.10: Algorithm of updating process of "Security Principles" database

Once the information that differs between the versions of the catalogues have been identified, they can be manually allocated to the categories they fit in better. The automated part of this process is completed in seconds, while the manual section depends on the number of new or updated entries.

3.7 Methodology in ResilBlockly

This methodology can be applied to ResilBlockly partially or as a whole. The connection can be identified in three levels:

- The modelling phase divides the system into components with the use of “Classes” and provides the user with all the information they require to determine the technical requirements and security configurations.
- As the selection of the weaknesses and the vulnerabilities is based on online security catalogues, the categorisation of this information, based on the relationships established among the catalogues and their connections to security standards like the ISA/IEC 62443 Standard, can provide an automated filtering phase based on the characteristics of the modelled systems.
- Lastly, considering the number of threats that a system composed of several components may contain, a graphical representation of the weaknesses, attack patterns, and vulnerabilities identified will provide a deeper understanding for the end-user and the opportunity for the conductor of the assessment to verify that all the information is relevant and complete.

4 Testing and result analysis

This section is devoted to the testing of the designed methodology seen in chapter 3. The testing is applied on the ICT Gateway use case and the updating process from one version of catalogues to the other. After the testing is completed, the presented results are analysed and an evaluation of the methodology is provided, highlighting the successful points and the issues that arose.

4.1 Use case: ICT Gateway

The ICT Gateway is the use case chosen to test the designed methodology. Taking advantage of the fact that every component of the ICT Gateway is considered an individual system, the risk assessment is deployed on the assets of the latter by analysing their characteristics and functionalities.

4.1.1 Introduction

ICT Gateway is a medium among data collection actuation subsystems and domain operations in the context of Smart Grids. The system provides a supportive environment for data, configuration, and control flows [67].

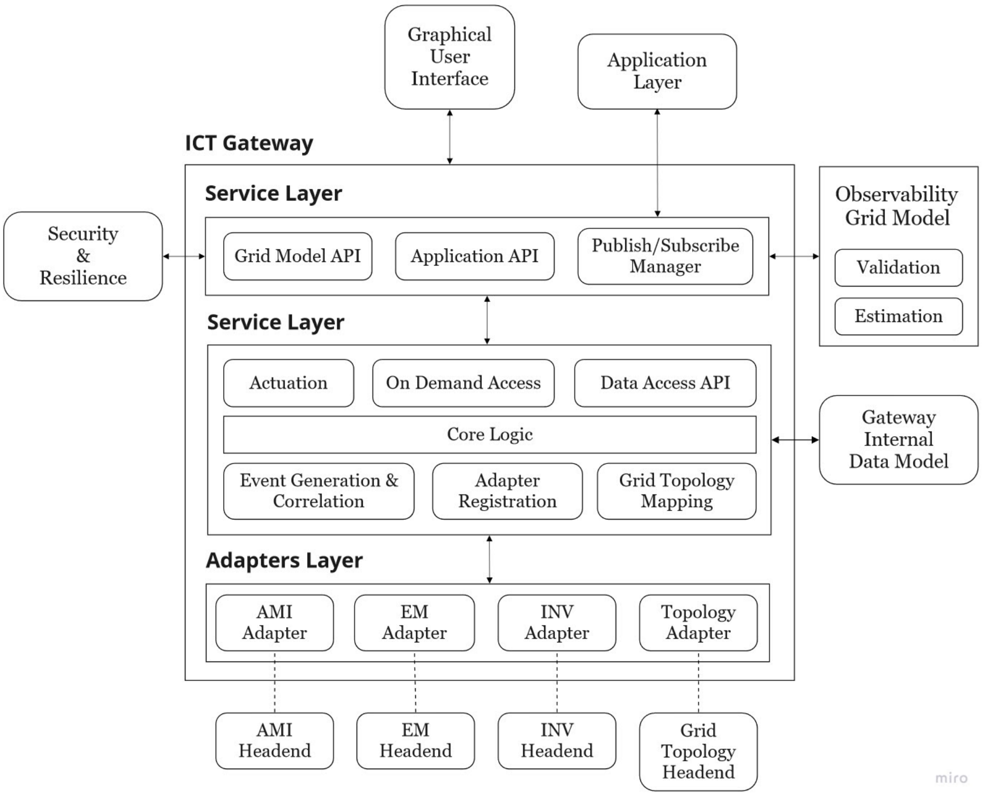


Figure 4.1: Architecture of ICT Gateway [68]

Figure 4.1 presents a “high-level architectural description” where the interacting subcomponents are positioned inside and outside the ICT Gateway environment [68].

In this thesis, the testing of the designed methodology is presented on two components identified in the ICT Gateway system. The chosen components are the “Graphical User Interface” and the “Security & Resilience”.

Table 4.1 lists the technical characteristics of the components determined as the “programming languages”, the “communication protocols”, the “security configurations”, the “libraries”, and the “interfaces”.

Table 4.1: Technical Characteristics of ICT Gateway components

Technical Characteristics	Graphical User Interface	Security & Resilience
Programming Languages	TypeScript (Angular 8 Framework)	Java
Communication Protocols	HTTP MQTT	HTTP MQTT
Interfaces	HTTP REST API MQTT Broker MySQL Connector Bridge Object-Relational Mapping (ORM) Hibernate	HTTP REST API MQTT Broker
Libraries	Bootstrap 4	
Security Configurations	Sec-3, Sec-4	Sec-3, Sec-4

In order to avoid any security breaches, the weaknesses, attack patterns and vulnerabilities explored in this thesis are connected to the assumption that adequately implemented security requirements [68] are not present. More specifically, it is assumed that ICT Gateway lacks the following security measures:

- Sec-03, that requires any internal communication to be protected by authentication and encryption mechanisms
- Sec-09, that requires ICT Gateway to detection mechanisms that ensure the integrity of the managed data

4.1.2 Component 1: Graphical User Interface

The “Graphical User Interface (GUI)” [68] component is a means of interaction between the “Distribution System Operators (DSO)” and the system. GUI offers input and visual output for ICT Gateway processes. GUI communicates with other components through Message Queuing Telemetry Transport (MQTT) and HTTP Representational State Transfer (REST) API. MQTT is a “publish/subscribe lightweight messaging protocol” [69], and HTTP REST API is a web server linking a client to a system’s information and processes based on the technical description of the World Wide Web’s functionality [70]. The DSO and the ICT Gateway reach GUI through HTTP REST API, and it obtains updates of the status of ICT Gateway through the MQTT Broker.

The testing of the methodology begins by analysing the target system and manually identifying its components. Considering the GUI as the target system, the identified technical characteristics seen in Table 4.1 represent the components, also known as assets. After the components have been analysed, the filtering process of the weaknesses and the attack patterns is carried away through the security principles. Hence, the FRs are manually selected for the GUI as a whole from the “CWE Security Principles” database, followed by selecting the SRs for the separate components.

Table 4.2 presents the identified FRs of the GUI as a whole, and Table 4.3 presents the identified SRs for each of the GUI components.

Table 4.2: FRs filtering of the GUI system

Foundational Requirements	**Technical Control System Requirements**
Access Control	Software Process and Device Identification and Authentication
Use Control	Authorization Enforcement
Data Confidentiality	Use of Cryptography
Data Integrity	Communication Integrity Input Validation Deterministic Output Software and Information Integrity

Table 4.3: SRs filtering of the GUI system

Technical Control		HTTP	MQTT	
System Requirements	**Typescript**	**REST API**	**Broker**	**ORM**
Software Process and Device Identification and Authentication	X	X	X	
Authorization Enforcement	X	X	X	
Use of Cryptography	X	X	X	
Communication Integrity	X	X	X	
Input Validation	X			X
Deterministic Output	X			
Software and Information Integrity	X			

Hence, the programming language (Typescript) is used to filter all the weaknesses identified for GUI, the SRs of authentication, authorization and encryption are linked to MQTT Broker and HTTP REST API, input validation is linked to ORM while the functionalities of GUI filter the rest. After the FRs and SRs are selected, the security principles for every abstraction level are filtered, and the relevance of the resulting weaknesses is evaluated through the use of keywords extracted from the technical characteristics and GUI and its functionality, some of which are: HTTP, HTTP REST, MQTT, ORM and Hibernate. The sub-principles are manually filtered, providing a set of relevant weaknesses while the filtering by keywords is done semi-automatically by extracting from the "CWE Security Principles" database only the weaknesses that contain the keywords.

Table 4.4 presents the selected security principles and examples weaknesses identified in the process. Each level of the sub-principles contains several weaknesses, and they are filtered by the component they are connected to in Table 4.3 or the descriptions of their functionalities.

As Table 4.4 represents the potential weaknesses of GUI based on its description, the last part of this phase is to remove the already mitigated ones, which is semi-automated mechanism as the manual selection of implemented mitigations leads to the automatic removal of the weaknesses it affects. The process is repeated for the attack patterns following the same steps, and after the relationships between the related weaknesses and attack patterns have been established, all the relevant information is stored in a database.

Table 4.4: Weakness filtering of the GUI system

Security Sub-Principles		Filtered Weaknesses
Communications Channel Authentication	Double-sided Authentication	CWE-300
Authentication Framework/Library		CWE-307
Source Authentication		CWE-441
Access Control Authorization	Permissions Assignment	CWE-77
Principles of Least Privilege		CWE-89
Separation of Privileges		CWE-15
Transited Data Encryption	Communication Channel Encryption	CWE-319
Cryptographic Algorithm Management		CWE-327
Protected Communication		n/a
Input Validation	Allowlists and Denylists	n/a
User-controlled Input Validation		n/a
Output Encoding	Allowlists and Denylists	n/a
	Consistent Output Encodings	n/a

The process of identifying the vulnerabilities of GUI is completed by utilizing the information of the employed components. In this case, the focus will be on the “Bootstrap 4” library and the “Object-Relational Mapping (ORM) Hibernate” interface. To avoid jeopardizing the security of the system, random versions are chosen to apply the methodology. Based on the information extracted by CPE, the following are provided as input to generate the related CVE vulnerabilities:

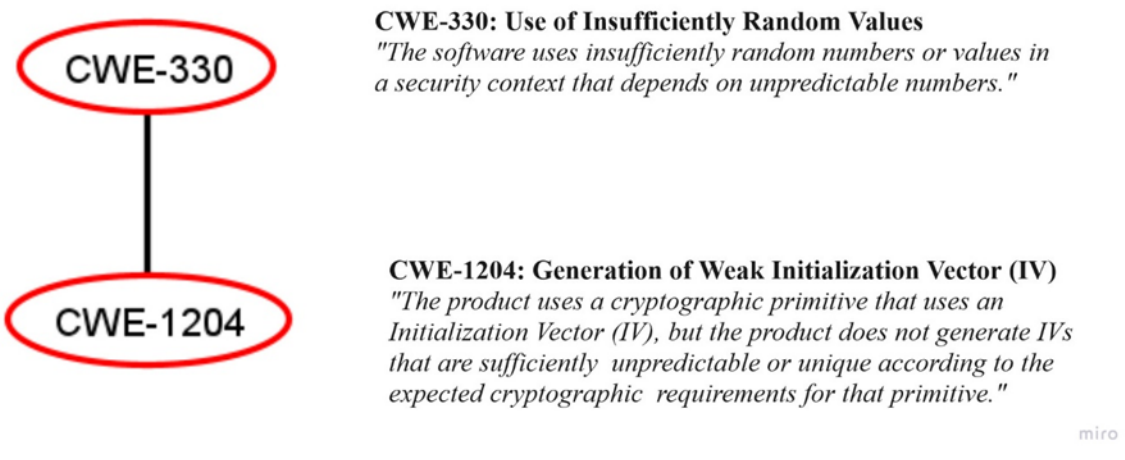
- Vendor: Hibernate / GetBootstrap
- Product: Hibernate ORM / Bootstrap
- Version: 3.6.0 / 4.0.0
- Update: Beta 4 / Alpha

Hence, the user sets the above information as a request to the “CVE-CPE” database, which maps the CPE characteristics to the CVE vulnerabilities through NVD, and the list of relevant CVE vulnerabilities is generated automatically. After the CVE vulnerabilities are extracted, they are linked to the related CWE weaknesses through NVD and then combined with the ones extracted from the CWE catalogue. This process is done automatically using Talend which maps the extracted information to the stored data in the "NVD vulnerabilities" and "CWE weaknesses" databases. Table 4.5 provides the collected results and CVSS severity levels found in NVD.

Table 4.5: Vulnerabilities of GUI

Employed Catalogues	Weaknesses	Vulnerabilities	CVSS (v.3, v.2)
CPE, NVD	CWE-89	CVE-2020-25638	7.4, 5.8
CPE, NVD		CVE-2019-14900	6.5, 4.0
CPE, NVD	CWE-79	CVE-2018-14042	6.1, 4.3
CPE, NVD		CVE-2018-14041	6.1, 4.3
CPE, NVD		CVE-2018-14040	6.1, 4.3

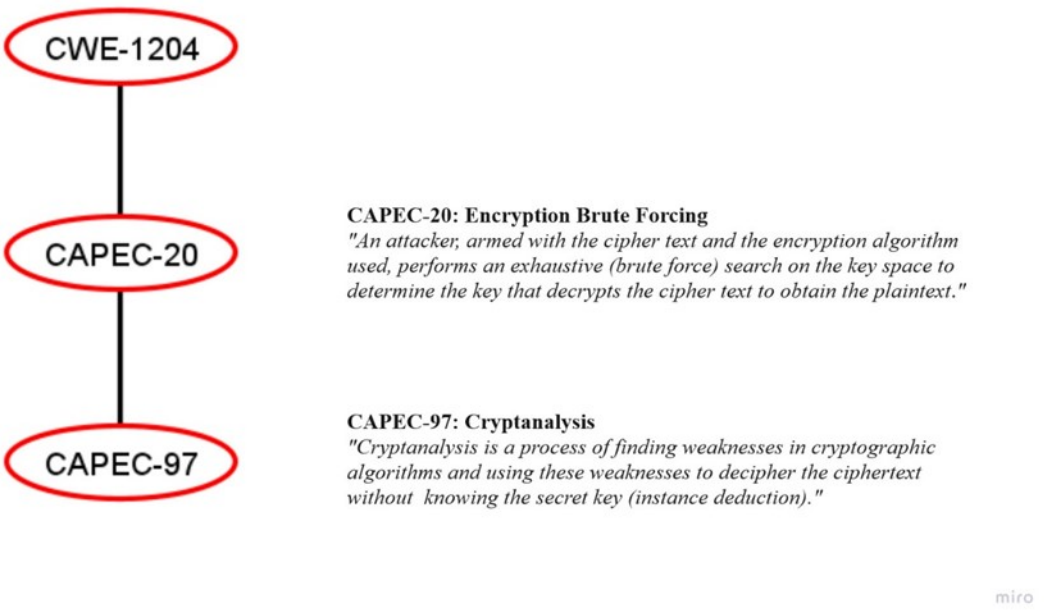
Once all the information is collected, the graphical representations of their relationships are automatically generated using Talend for the XML file and ADTool for the graphical trees. Figure 4.2, 4.3 and 4.4 present the "Hierarchical Weakness Tree" for "`CWE-330: Use of Insufficiently Random Values`" [71], the "Attack Path Tree" leading to "`CWE-1204: Generation of Weak Initialization Vector (IV)`" [72], which is a leaf of the previous tree, and the respective "Attack-Defense Tree". Figure 4.5 presents the "Weakness-Vulnerability Tree" for the "`CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')`" [73] weakness of the "Hibernate ORM".



CWE-330: Use of Insufficiently Random Values
"The software uses insufficiently random numbers or values in a security context that depends on unpredictable numbers."

CWE-1204: Generation of Weak Initialization Vector (IV)
"The product uses a cryptographic primitive that uses an Initialization Vector (IV), but the product does not generate IVs that are sufficiently unpredictable or unique according to the expected cryptographic requirements for that primitive."

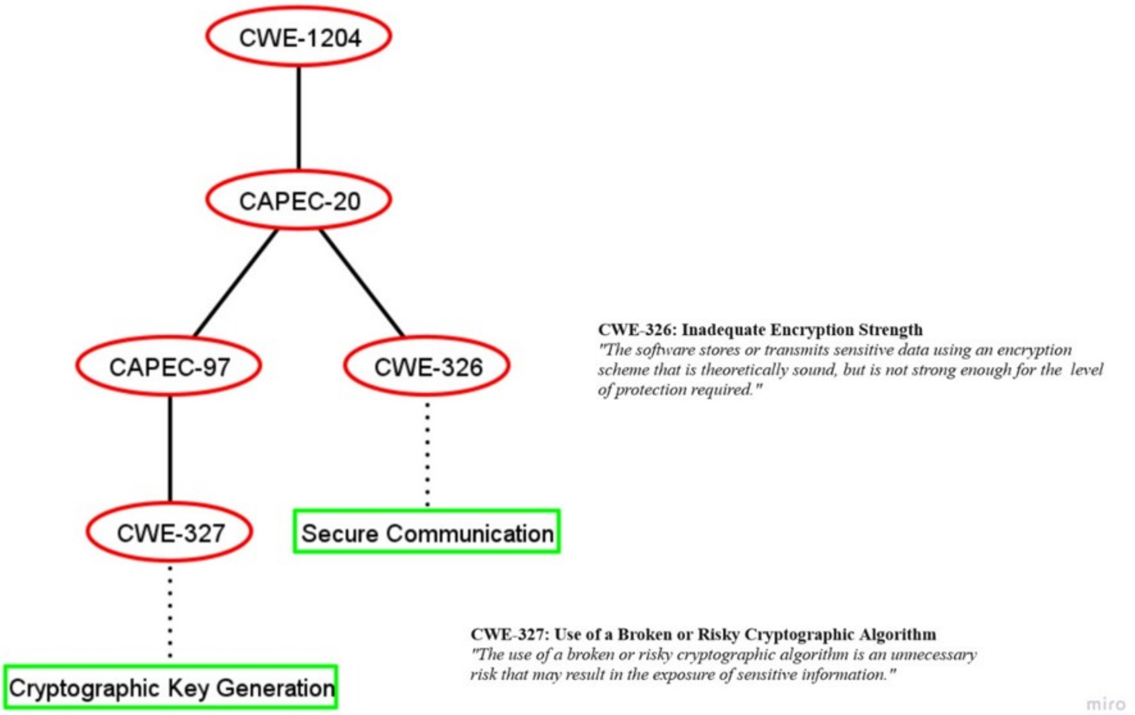
Figure 4.2: "Hierarchical Weakness Tree" for "CWE-330"



CAPEC-20: Encryption Brute Forcing
"An attacker, armed with the cipher text and the encryption algorithm used, performs an exhaustive (brute force) search on the key space to determine the key that decrypts the cipher text to obtain the plaintext."

CAPEC-97: Cryptanalysis
"Cryptanalysis is a process of finding weaknesses in cryptographic algorithms and using these weaknesses to decipher the ciphertext without knowing the secret key (instance deduction)."

Figure 4.3: “Attack Path Tree” leading to “CWE-1204”



CWE-326: Inadequate Encryption Strength
"The software stores or transmits sensitive data using an encryption scheme that is theoretically sound, but is not strong enough for the level of protection required."

CWE-327: Use of a Broken or Risky Cryptographic Algorithm
"The use of a broken or risky cryptographic algorithm is an unnecessary risk that may result in the exposure of sensitive information."

Figure 4.4: “Attack-Defense Tree” of “CWE-1204”

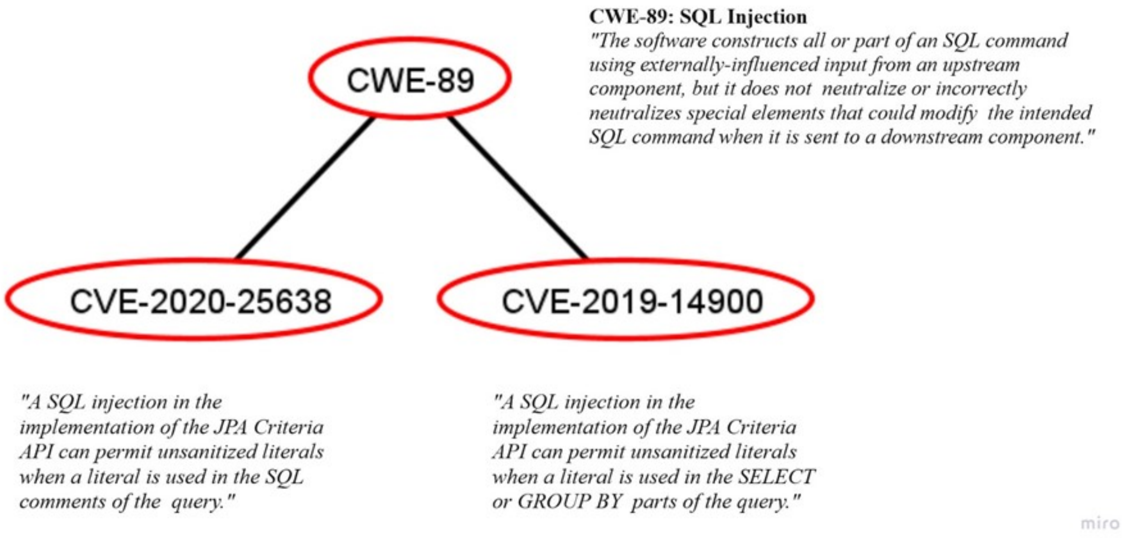

CWE-89: SQL Injection
"The software constructs all or part of an SQL command using externally-influenced input from an upstream component, but it does not neutralize or incorrectly neutralizes special elements that could modify the intended SQL command when it is sent to a downstream component."
CWE-89
CVE-2020-25638
CVE-2019-14900
"A SQL injection in the implementation of the JPA Criteria API can permit unsanitized literals when a literal is used in the SQL comments of the query."
"A SQL injection in the implementation of the JPA Criteria API can permit unsanitized literals when a literal is used in the SELECT or GROUP BY parts of the query."
miro

Figure 4.5: “Weakness-Vulnerability Tree” of “CWE-89”

4.1.3 Component 2: Security and Resilience

The “Security and Resilience (S&R)” [68] component is a software module, individually positioned outside of the ICT Gateway frame, whose aim is to supervise the system and identify any functionality issues that may occur. S&R collaborates with the “Event Generation & Correlation (EGC)” to establish a safe, resilient, and robust environment through detection processes of faults and attacks. S&R communicates with other components through MQTT and HTTP REST API. S&R functions by alerting the DSO on GUI through MQTT and extracting the identified anomalies from EGC and the “Data Access API” through HTTP REST API.

The exact process is repeated for the S&R, so considering it as the target system, the identified technical characteristics in Table 4.1 represent the components. After the components have been manually analysed, the filtering process of the weaknesses and the attack patterns is carried away through the security principles. Hence, the FRs are selected for the S&R as a whole, followed by selecting the SRs for the separate components.

Table 4.6 presents the identified FRs of the S&R as a whole, and Table 4.7 presents the identified SRs linked to the components of S&R.

Table 4.6: FRs filtering of the S&R system

Foundational Requirements	Technical Control System Requirements
Access Control	Software Process and Device Identification and Authentication
Use Control	Authorization Enforcement
Data Confidentiality	Use of Cryptography
Data integrity	Communication Integrity Error Handling Security Functionality Verification
Restrict Data Flow	Zone Boundary Protection
Timely Response to Event	Continuous Monitoring

Table 4.7: SRs filtering of the S&R system

Technical Control System Requirements	Java	HTTP REST API	MQTT Broker
Software Process and Device Identification and Authentication	X	X	X
Authorization Enforcement	X	X	X
Use of Cryptography	X	X	X
Communication Integrity	X	X	X
Error Handling	X		
Security Functionality Verification	X		
Zone Boundary Protection	X		
Continuous Monitoring	X		

Hence, the programming language (Java) is used to filter all the weaknesses identified for S&R, the SRs of authentication, authorization and encryption are linked to MQTT Broker and HTTP REST API, while the rest are filtered by the functionalities of S&R. After the FRs and SRs are manually selected, the security principles for every abstraction level are filtered, and the relevance of the resulting weaknesses is semi-automatically evaluated through the use of keywords extracted from the technical characteristics and S&R and its functionality, some of which are: HTTP, HTTP REST, and MQTT.

Table 4.8 presents a subset of the selected security principles and examples of weaknesses identified in the process. Each level of the sub-principles contains several weaknesses, and they are filtered by the component they are connected to in Table 4.1 or the descriptions of their functionalities.

Table 4.8: Weakness filtering of the S&R

Security Sub-Principles		**Filtered Weaknesses**
Communications Channel Authentication	Double-sided Authentication	CWE-300
Authentication Framework/Library		CWE-307
Source Authentication		CWE-441
Access Control Authorization	Permissions Assignment	CWE-77
Principles of Least Privilege		CWE-89
Separation of Privileges		CWE-15
Transited Data Encryption	Communication Channel Encryption	CWE-319
Cryptographic Algorithm Management		CWE-327
Protected Communication		n/a
Error Detection	Warnings	n/a
Error Messages	Default Error Messages	n/a
Configuration Evaluation		n/a
Protocols Evaluation		n/a
Software Zone Separation		n/a
Attack Surface Reduction		n/a
Behavior Monitoring		n/a
Certificate Monitoring		

As Table 4.8 represents the potential weaknesses of S&R based on its description, the last part of this phase is to semi-automatically remove the already mitigated ones. The process is repeated for the attack patterns following the same steps and after the relationships between the related weaknesses and attack patterns have been established, all the relevant information is stored in a database.

The process of identifying the vulnerabilities of S&R is completed by utilizing the information of the employed components. In this case, the focus will be on the chosen MQTT Broker. To avoid jeopardizing the security of the system, the "Eclipse Mosquitto" broker is chosen at random to apply the methodology. Based on the information extracted by CPE, the following are provided as input to generate the related CVE vulnerabilities:

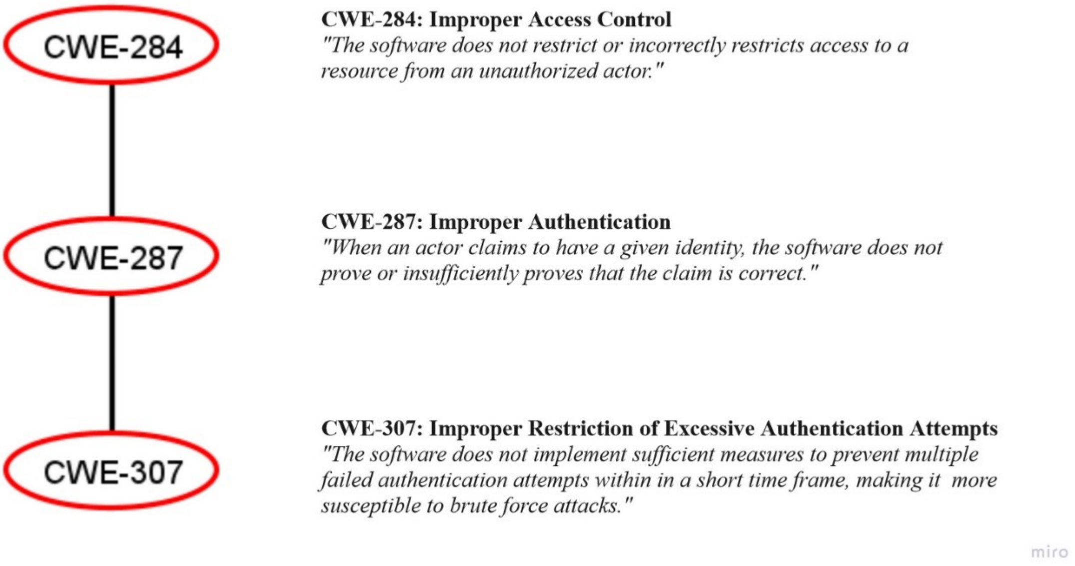
- Vendor: Eclipse
- Product: Mosquitto
- Version: 1.5

After the CVE vulnerabilities are semi-automatically extracted using CPE, they are linked automatically to the related CWE weaknesses through NVD and then combined with those extracted from the CWE catalogue, using Talend. Table 4.9 provides the collected results and CVSS severity levels found in NVD.

Table 4.9: Vulnerabilities of S&R

Employed Catalogues	Weaknesses	Vulnerabilities	CVSS (v.3, v.2)
CPE, NVD	CWE-754	CVE-2019-11779	6.5, 4.0
CPE, NVD	CWE-287	CVE-2018-12551	8.1, 6.8
CPE, NVD	CWE-440	CVE-2018-12550	8.1, 6.8
CPE, NVD	CWE-284	CVE-2018-12546	6.5, 4.0
CPE, NVD	CWE-732	CVE-2018-20145	7.5, 5.0
CVE, NVD	CWE-20	CVE-2019-5432	7.5, 5.0

Once all the information is collected, the graphical representations of their relationships are automatically generated using Talend for the XML file and ADTool for the graphical trees. Figure 4.6, 4.7 and 4.8 present the "Hierarchical Weakness Tree" for "`CWE-284: Improper Access Control`" [74], the "Attack Path Tree" leading to "`CWE-307: Improper Restriction of Excessive Authentication Attempts`" [75], which is a leaf of the previous tree, and the respective "Attack-Defense Tree". Figure 4.9 presents the "Weakness-Vulnerability Tree" for the "`CVE-2019-11779`" [76] vulnerability of the "Eclipse Mosquitto" MQTT Broker.

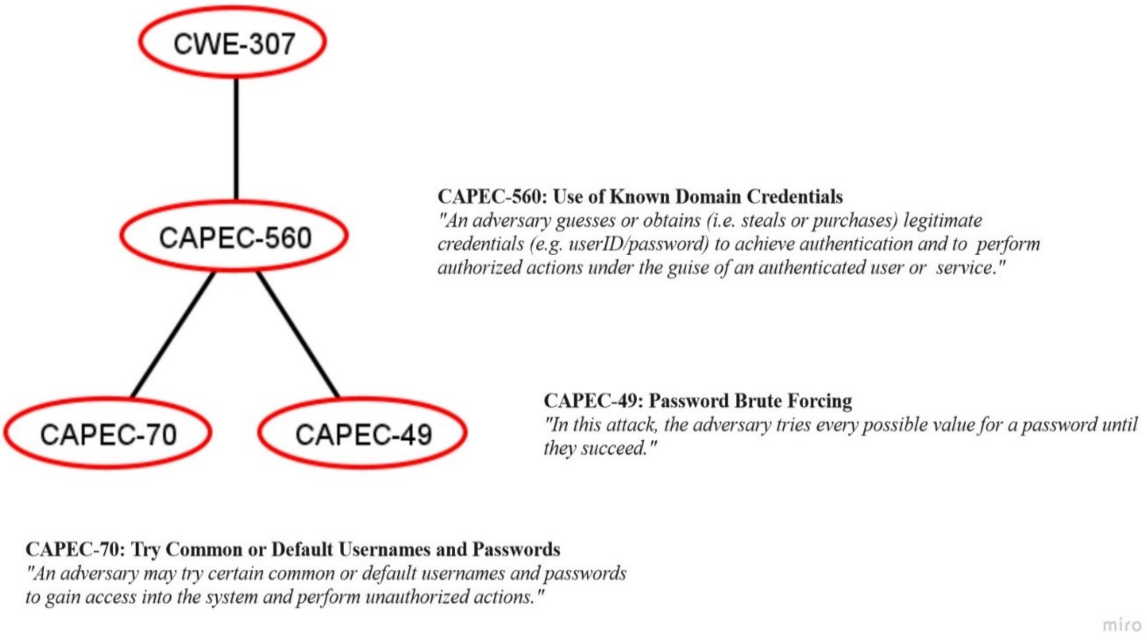


CWE-284: Improper Access Control
"The software does not restrict or incorrectly restricts access to a resource from an unauthorized actor."

CWE-287: Improper Authentication
"When an actor claims to have a given identity, the software does not prove or insufficiently proves that the claim is correct."

CWE-307: Improper Restriction of Excessive Authentication Attempts
"The software does not implement sufficient measures to prevent multiple failed authentication attempts within in a short time frame, making it more susceptible to brute force attacks."

Figure 4.6: “Hierarchical Weakness Tree” for “CWE-284”

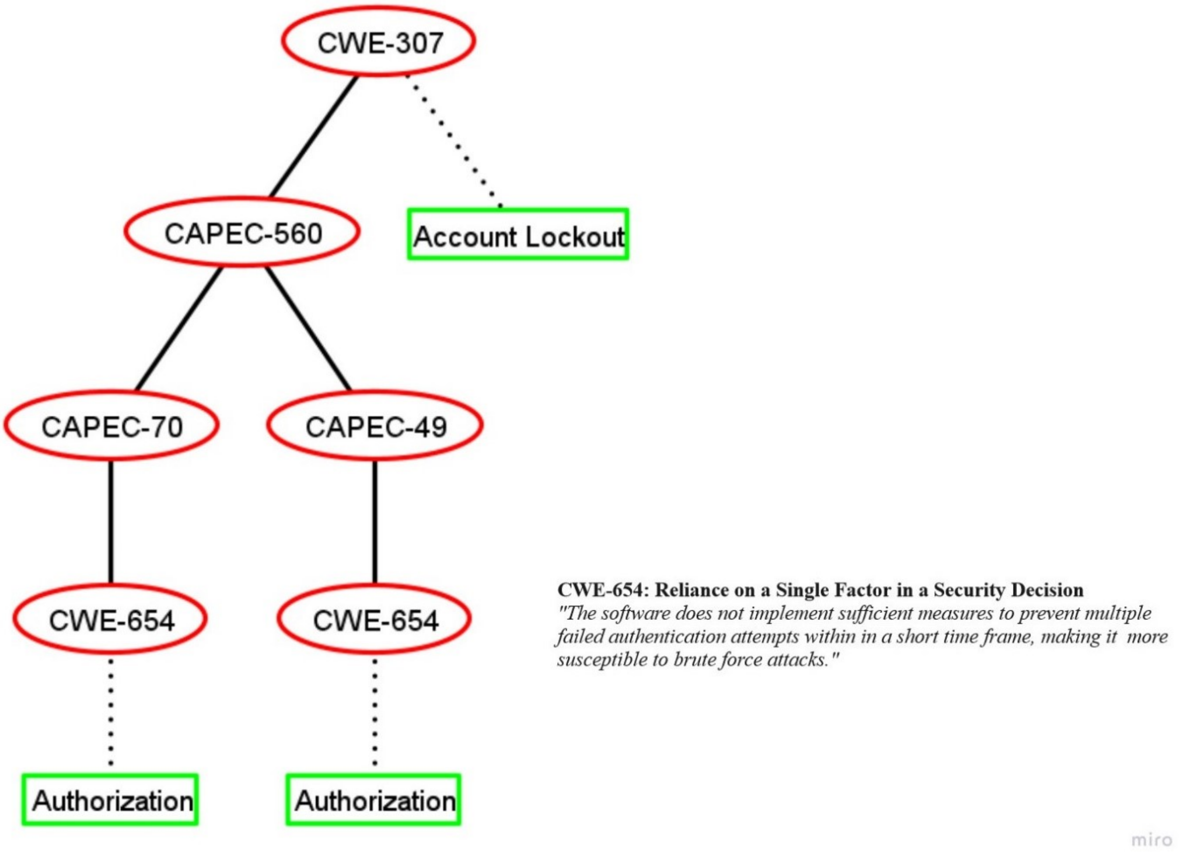


CAPEC-560: Use of Known Domain Credentials
"An adversary guesses or obtains (i.e. steals or purchases) legitimate credentials (e.g. userID/password) to achieve authentication and to perform authorized actions under the guise of an authenticated user or service."

CAPEC-49: Password Brute Forcing
"In this attack, the adversary tries every possible value for a password until they succeed."

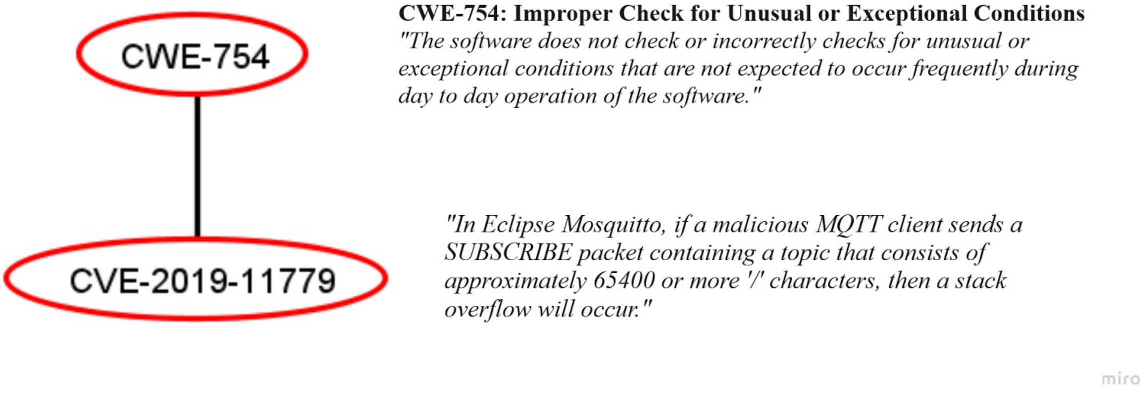
CAPEC-70: Try Common or Default Usernames and Passwords
"An adversary may try certain common or default usernames and passwords to gain access into the system and perform unauthorized actions."

Figure 4.7: “Attack Path Tree” leading to “CWE-307”



CWE-654: Reliance on a Single Factor in a Security Decision
"The software does not implement sufficient measures to prevent multiple failed authentication attempts within in a short time frame, making it more susceptible to brute force attacks."

Figure 4.8: “Attack-Defense Tree” of “CWE-307”



CWE-754: Improper Check for Unusual or Exceptional Conditions
"The software does not check or incorrectly checks for unusual or exceptional conditions that are not expected to occur frequently during day to day operation of the software."

"In Eclipse Mosquitto, if a malicious MQTT client sends a SUBSCRIBE packet containing a topic that consists of approximately 65400 or more '/' characters, then a stack overflow will occur."

Figure 4.9: “Weakness-Vulnerability Tree” of “CVE-2019-11779”

4.2 Updating process

The testing process of the update is also divided into two segments and applied in different versions of the CWE catalogue. For CWE, version 4.3 was used as the “old” version, and version 4.4 was used as the “new and updated” version. The first part is implemented solely in Talend. Version 4.4 for CWE is downloaded in XML format and used as input in Talend in order to generate the “Hierarchical Weaknesses” databases, the databases containing the “ParentOf’-” ChildOf” connections between the two repositories, in CSV format.

In order to test the updating process of the “CWE Security Principles” database, the updating algorithm seen in Figure 3.10 was implemented using Python in Jupyter Notebook. In the case of the CWE, the Python code generates a list of the weaknesses present in both versions with an automatic link to the security principles and a list of the new and updated weaknesses only present in CWE version 4.4. The outcome provides twenty-six new entries for twenty individual weaknesses, which are manually linked to security principles.
Table 4.10 presents the weaknesses and their security principles’ categories.

Table 4.10: Number of categories and weaknesses

Foundational Requirements	**Technical Control System Requirements**	**Security Sub-principles**	**Weaknesses ID**
Data integrity	Error Handling	Error Detection	134
	Soft. & Info. Integrity	Type Evaluation Object Comparison	1236, 131, 134, 194, 597, 469, 481
	Input Validation	Numeric Input Validation Variable Initialization	124, 1333, 329, 190
Data Confidentiality	Sec. Funct. Verification	Security Locks Eval.	366
	Use of Cryptography	Rand. Gen. Initial Values	1204
Restrict Data Flow	Application Partitioning	Data Partitioning	20
Timely Response to Event	Continuous Monitoring	Behavior Monitoring Signal Handlers	1281, 1332, 364, 674

The information in Table 4.10 represents new weaknesses or existing weaknesses whose mitigations have been updated. For instance, “`CWE-131: Incorrect Calculation of Buffer Size`” [77] is an updated weakness, where the mitigation “Use the appropriate type for the desired action. For example, in C/C++, only use unsigned types for values that could never be negative, such as height, width, or other numbers related to quantity. This will simplify validation and will reduce surprises related to unexpected casting.” is added in the new version.

Weakness "`CWE-1333: Inefficient Regular Expression Complexity`" [78] is a new addition altogether.

4.3 Result analysis

Considering the application of the designed methodology on the components of the ICT Gateway in sections 4.1.2 and 4.1.3, an analysis is performed on the results recovered.

During the identification process, a common characteristic for the weaknesses and the attack patterns was that the filtering mechanisms in place significantly limited the number of relevant entries. More specifically, for "authentication", the CWE catalogue provides 680 relevant results. As authentication and authorization are concepts that are considered in the CWE search, the equivalent number of related weaknesses in the "Access Control" and "Use Control" FRs is 224, which after selecting relevant SRs and sub-principles is reduced to 146 where the keyword search is applied.

The system analysis exposed the existence of shared technical requirements and security configurations among the components of the system. Either partially or as a whole, some aspects of the analysed system are repeated across the components that compose it. For example, both the tested components use "MQTT Broker" and "HTTP REST API", whose communications are protected by authentication and encryption mechanisms under "Sec-03". As their functionality is the same on both components, the identified weaknesses, attack patterns, and vulnerabilities are the same.

In extracting relevant vulnerabilities, the results highlight that the amount of information of the production of the hardware and software employed, controls the level of relevance of the vulnerabilities. Considering the number of available vulnerabilities and the multiple versions of the same affected product, a single piece of information can provide multiple results, out of which only a few are related to the scenario at hand. For instance, the "MQTT" would provide fifty-three results in CVE, only six of which are linked to "Eclipse Mosquitto 1.5".

Analysing the results extracted while connecting weaknesses and vulnerabilities from the CWE, CVE and NVD catalogues, it is evident that although CWE presents a connection between its weaknesses and the vulnerabilities in CVE, CVE does not offer such a relationship. Furthermore, NVD lists weaknesses found in CWE as relevant for the vulnerabilities it describes, but these relationships are not compliant with CWE. The "`CVE-2018-20145`" [79] vulnerability is linked to “`CWE-732: Incorrect Permission Assignment for Critical Resource`” [80] in the NVD catalogue, but the relationship is not present in the CWE catalogue.

5 Discussion

This section provides an in-depth analysis of the findings collected from the drafting process of the methodology and the results generated from the use case. The outcomes evaluated and linked to the relevant literature presented in section 2.3, their impact outside the scope of the thesis is highlighted, and an insight is given in further related research topics.

5.1 Research gap

During the drafting of the methodology presented in chapter 3, different features of the process indicated notable results.

Firstly, using the ISA/IEC 62443 standard in the risk assessment to link the assets to the weaknesses and attack patterns provides a structured mapping between the technical requirements of the standard and the system. The highlight of this approach is the interpretation of the technical requirements as security principles providing a potential compliance validation method. The compliance of systems to standards, although necessary, is described as a complicated and expensive process. According to Portela et al. [33], implementing two security standards, ISA/IEC 62443 and ISO/ISA 27001, would not be recommended due to the high maintenance price. Hence, the inclusion of the technical perspective of the risk assessment process in an underlying manner partially simplifies the application and maintenance of standard compliance.

Secondly, after the security principles have filtered the relevant weaknesses and attack patterns of the assets, more in-depth filtering is carried out using the technical characteristics of the components of each asset. These characteristics filter the weaknesses and the attack patterns by verifying their presence in the descriptive categories of the CWE and CAPEC catalogues seen in the semantic approach presented by A.Brazhuk [31].

Lastly, the use of tree-based graphical representations of the connections among weaknesses, vulnerabilities and attack patterns offers the ideal presentation of the relationships between the information of the catalogues. The approach in this thesis is a combination of the "security knowledge graph" generated by Xiao et al. [36] and "Attack-Defense Trees" seen in sections 2.3.6 and 2.3.7, which simplify the understanding of the numerous connections between the information and the catalogues from the "path" format they embody.

5.2 Results evaluation

Based on the analysis of the results, the advantages and disadvantages of applying the designed methodology are listed, followed by an evaluation of the methodology as a whole across the objectives set in section 1.2.

On the one hand, this methodology limits the number of weaknesses, attack patterns, and vulnerabilities the user is required to manually filter for relevance to the components under assessment and their technical and security requirements, as every selection determines the list of relevant entries. Additionally, the selection process is more user-friendly as the user is asked to choose the system's requirements instead of selecting the threats it might face.

The existence of repeated functions and components minimises the assessment process as technical characteristics, and their security features are required to undergo assessment only once, and their results can be applied in more than components they comply with. Even in the case that the similarity is partial, a "base model" can be generated on which new attributes can be added to specialise the results according to the given component.

On the other hand, the methodology bases the extraction of security information on the components of the system, the technical characteristics, which include a thorough description of the versions, updates and vendors of the hardware and software in use, and the proper definition of the components. This feature diminishes the stability of the effort required and the quality of the results acquired based on their relevance to the scenario. Also, as the methodology relies so heavily on the information available on the online catalogue, it is notable that although the entries are reliable, the catalogues are not complete, nor are the established relationships. Hence, it is the user's responsibility during the collection of the information to verify that no information is missing or manually add relevant data. This result was expected considering the work seen in section 2.3.4 by Xiao et al. [36], where part of the "security knowledge graphs" is dedicated to the "missing" connections. In the methodology seen in section 3, the consequences of "missing" relationships are not crucial, as the extracting is applied on all the catalogues; hence, the Xiao et al. [36] research validates the graphical representations of the connections as a means to correctly linking the collected weaknesses, vulnerabilities, and attack patterns to targeted assets. More specifically, as all the identified security concepts and their connections are presented to the user as tree-like graphs, the user can easily manipulate the connections to determine new paths, establishing the new relationships.

Therefore, the methodology provides a complete path of the risk assessment from identifying any potential threats on the system to the graphical representation of the latter accompanied by the steps that lead to exploitation and the proposed mitigations. However, such a methodology requires an extensive understanding of the technical characteristics and the system's functionalities, making the identification process of the threats overly dependable on the amount of information the user holds and the understanding and interpreting the categories provided. So, the successful implementation of the designed methodology depends on the environment of use and the technical level of the user applying it to the system. This outcome is closely connected to the work seen by Franqueira et al. [39] in section 2.3.5, as the argumentation-based approach of RSA is equivalent to the understanding that the user holds. Both approaches highlight that since the security concepts used in filtering are not standard options, they can simplify the undertaken process or create added risky security gaps. Furthermore, RSA emphasises identifying assets and functional and security system requirements before the argumentation input. This setting verifies the assumption seen from the results in section 4.3, where extensive knowledge of the system, its components and the required security conditions is vital for the entire risk assessment process.

5.3 Impact

The concept analysed in this thesis and the results it generated have a significant impact outside the scope of the work carried out in this thesis in two levels: 1) contribution to the existing research and approaches on the topics managed, and 2) the simplification of the risk assessment process on IT/OT systems.

By using as a basis of the existing work the literature introduced in section 2.3, the approach and results that were seen in this thesis expand on the current research by:

- Using the technical descriptions of the assets as the characteristics connecting them to the semantic approach in the work of A.Brazhuk [31] and the ontology approach in the work of Wang et al. [35],
- Expanding the notion of "Attack-Defense Trees" used on the information of the online catalogues presented in the section 2.3.6 and 2.3.7, by exploiting the "CanFollow-CanPrecede" relationship and offering a path from the attack to the weakness and its mitigation, and
- Combining the information found in NVD to add the missing connections between CVE vulnerabilities and CWE weaknesses, which is identified in the work of Xiao et al. [36].

Furthermore, new perspectives are introduced that deviate from the work seen in literature:

- The role of the ISA/IEC 62443 standard in this thesis is based on the part of the proposed technical requirements and is in direct connection to the weaknesses and attack patterns of the CWE and CAPEC catalogues, and
- The use of the ISA/IEC 62443 standard as the starting point of the risk assessment methodology and as a link between the assets and their security threats provides an indirect evaluation of the system's compliance with the system.

Moreover, assessing the security condition of IT/OT systems and evaluating the probability that any security openings will be exploited has numerous restrictions. On one side, such an assessment would require extensive access to security-sensitive hardware and software, and on the other side, any testing action would significantly limit the availability of the system. Hence, the ability to list any weaknesses, vulnerabilities, and attack patterns relevant to a system without testing the actual system would impact the overall approach to the risk assessment process of such systems. This concept is more wholesome when the methodology is applied with ResilBlockly, as the modelling of the system provides a simulation of the system, its components and their functionalities making it easier to assess if the weaknesses are relevant to the given scenario and if the selected attack paths can successfully exploit them.

Hence, the impact of this approach offers an assessment methodology that does not affect the availability of the services provided by the IT/OT systems but requires an in-depth understanding of the systems' functionality, which might require trust between the involved parties.

5.4 Future applications

The work completed in this thesis has multiple areas of further study that can expand the range of the offered applications.

Firstly, different systems might require compliance to different security standards, such as ISA TR84.00.09 – Security Related to Safety Instrumented Systems (SIS). Therefore, categorizing the weaknesses and attack patterns of the CWE and CAPEC catalogues into "security principles" that are extracted from the necessary standard can offer a dual functionality: risk assessment and compliance of the system to the required standards. The compliance evaluation process may easily be semi-automated and indirectly applied through the risk assessment. More specifically, by identifying the requirements that each standard sets for the system they target, the weaknesses and the attack patterns can be linked to these requirements through their mitigations. Hence a direct link is established between them. Once the standard is chosen for the risk assessment, following the presented methodology, the compliance is determined through the weaknesses identified by listing the vulnerable requirements of the standard according to the potential weaknesses that luck mitigations.

Another further application of the work seen in this thesis is the use the graphical trees to generate automated tests on the modelled system. These tests can be used as replacements for the penetration testing used to identify a system's weaknesses and evaluate if the selected attack patterns can lead to the identified weaknesses. This expansion of the methodology is based on the work by S. Salva and L. Regainia as presented in "An Approach for Guiding Developers in the Choice of Security Solutions and in the Generation of Concrete Test Cases" [81]. The approach takes advantage of the ADTool generated Attack-Defense Trees to create test cases through Eclipse, which are executed on the system and a list of results is provided indicating whether or not the attacks were successfully carried away. The test provides two types of results:

- if the attacks are successful, the weaknesses connected to these paths are verified as weak points, and
- once the mitigations are applied, the results will verify the successful application of the countermeasures

These tests are based on the "GWT" pattern:

- "Given": established the appropriate state for the action
- "When": initiates the action
- "Then": verifies the success or failure of the test case

In this case, the difference to be adapted would be the execution of the tests on the generated models in such a way that the results would reflect on the actual system. Hence, this format would offer the opportunity to test an IT/OT system without abstracting the availability of the functionalities offered by them.

6 Conclusion

The Industry's transition from 1.0 to 4.0 offered numerous advantages in the quality of performance and productivity, but it also introduced the cybersecurity threats of Information Technology (IT) to Operational Technology (OT) systems. Hence, a methodology is designed for the risk assessment of IT/OT systems by employing publicly available security catalogues that list the known weaknesses, attack patterns and vulnerabilities and their descriptive characteristics like the mitigations and detection methods. The purpose of this methodology is to provide a risk assessment approach that does not hinder OT availability and a filtering process for the significant amount of data in security catalogues.

The risk assessment process begins with analysing the system under assessment by dividing its assets into components characterised by security configurations and technical requirements. After the target components are identified, they are mapped to the potential threats that might render them vulnerable to attacks. This is achieved by employing the Common Weakness Enumeration (CWE), Common Attack Pattern Enumeration and Classification (CAPEC), Common Vulnerabilities and Exposures (CVE), National Vulnerability Database (NVD) and Common Platform Enumerations (CPE) catalogues, which offer information on weaknesses, attack patterns and vulnerabilities identified and publicly classified. The methodology uses the established "ParentOf"-"ChildOf" and "CanFollow"-"CanPrecede" relationships between the entries of the same catalogue, and "RelatedTo" relationships between entries of different catalogues, which are seen between CWE and CAPEC or CVE and CPE.

Furthermore, the ISA/IEC 62443 Standard and the concept of security principles are applied to categorise the information of the catalogues further and link them to the system under assessment. The ISA/IEC 62443 Standard provides security requirements in the form of "Foundational Requirements (FRs)" and "Technical Control System Requirements (SRs)" that create levels of filtering for both weaknesses found in CWE and attack patterns found in CAPEC. Hence, the FRs connect the components as a whole to the weaknesses and attack patterns, while the SRs and security sub-principles further limit the number of relevant connections.

Following the weaknesses and attack patterns, the vulnerabilities are identified and filtered through the CPE catalogue that utilises the name, vendor, version and update information of the software and hardware in the components to detect the CVE vulnerabilities that might affect them. If the detected CVE vulnerabilities are also found in NVD, it offers direct links between the entries in CWE and the entries in CVE while providing information on the severity levels of the vulnerabilities through the different versions of CVSS.

Once all the information is collected, linked and filtered, the methodology takes advantage of the connections to represent the threats of a system graphically. These representations highlight the weaknesses, attack patterns, vulnerabilities, security patterns, and relationships established among them in four trees of unique focus.

This methodology was developed with ResilBlockly as a target system, where the provided system and its components are divided into subcomponents over the modelling phase. Links to the online catalogues are already implemented to apply the filtering steps before the searching process based on the modelled components. Hence, considering the information collected, the user would benefit from a graphical representation of the connections to assist in the filtering process and provide an in-depth understanding of the potential threats, mitigations, and steps to reach them.

In order to test this methodology, the ICT Gateway system is assessed through its components. The outcomes of the testing process verify that the methodology offers advantages compared to manual or single-factor search mechanisms, as it restrains the number of weaknesses or attack patterns the user is required to review manually. So, the information of the catalogues is categorised based on the security requirements they target or the particular software or hardware in use, which might be recurrent throughout the components. However, the dependability of the information on community-based catalogues and the knowledge and understanding of the user over the system affects the stability of the methodology.

References

[1] R. Kour, “Cybersecurity Issues and Challenges in Industry 4.0,” Applications and Challenges of Maintenance and Safety Engineering in Industry 4.0, pp. 84–101, 2020, doi: 10.4018/978-1-7998-3904-0.ch005.

[2] A. Hahn, “Operational Technology and Information Technology in Industrial Control Systems,” in Cyber-security of SCADA and Other Industrial Control Systems, vol. 66, E. J. M. Colbert and A. Kott, Eds. Cham: Springer International Publishing, 2016, pp. 51–68. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-32125-7_4

[3] L. Monostori, “Cyber-physical Production Systems: Roots, Expectations and R&D Challenges,” Procedia CIRP, vol. 17, pp. 9–13, Jan. 2014, doi: 10.1016/j.procir.2014.03.115.

[4] “CWE - Common Weakness Enumeration”. https://cwe.mitre.org/index.html (accessed May 17, 2021).

[5] “CVE - Common Vulnerabilities and Exposures”. https://cve.mitre.org/index.html (accessed May 17, 2021).

[6] “NVD - National Vulnerability Database”. https://nvd.nist.gov/ (accessed May 17, 2021).

[7] “NVD - CPE”. https://nvd.nist.gov/products/cpe (accessed May 17, 2021).

[8] “CAPEC - Common Attack Pattern Enumeration and Classification (CAPECTM)”. https://capec.mitre.org/index.html (accessed May 17, 2021).

[9] “PrOTectME”. https://www.resiltech.com/index.php/projects/102-protectme#detail_proj (accessed Jun. 28, 2021).

[10] E. Schiavone et al., “BIECO Deliverable D6.1 - Blockly4SoS Model and Simulator”, Jun. 2021. [Online]. Available: www.bieco.org/public-deliverables/

[11] Security for Industrial Automation and Control Systems: Terminology, Concepts, and Models, ANSI/ISA-62443-1-1 (99.01.01), 2007. [Online]. Available: https://www.isa.org/products/isa-62443-1-1-2007-security-for-industrial-automat

[12] D. J. Bodeau and R. D. Graubart, “Cyber Resiliency Design Principles,” MITRE, Bedford , MA, USA, Technical Report MTR 170001, May 2017. Accessed: May 17, 2021. [Online].

Available: https://www.mitre.org/publications/technical-papers/cyber-resiliency-design-principles

[13] T. V. Benzel, C. E. Irvine, T. E. Levin, T. D. Nguyen, P. C. Clark, and G. Bhaskare, “Design principles for security,” Naval Postgraduate School, Monterey, CA, USA, Technical Report NPS-CS-05-010, 2005. Accessed: May 17, 2021. [Online]. Available: https://calhoun.nps.edu/handle/10945/510

[14] Security for Industrial Automation and Control Systems: System security requirements and security levels, ANSI/ISA-62443-3-3 (99.03.03), 2013. [Online]. Available: https://www.isa.org/products/ansi-isa-62443-3-3-99-03-03-2013-security-for-indu

[15] M. Schumacher, “Foundations of Security Patterns,” in Security Engineering with Patterns: Origins, Theoretical Models, and New Applications, 1st ed., vol. 2754. Berlin Heidelberg: Springer-Verlag, 2003, ch. 7, pp. 97-119. [Online]. Available: https://link.springer.com/book/10.1007/b11930

[16] M. Audinot and S. Pinchinat, “On the Soundness of Attack Trees,” in 2016 International Workshop on Graphical Models for Security, 2016, pp. 25–38, doi: 10.1007/978-3-319-46263-9_2.

[17] J. B. Hong, D. S. Kim, C.-J. Chung, and D. Huang, “A survey on the usability and practical applications of Graphical Security Models,” Computer Science Review, vol. 26, pp. 1–16, Nov. 2017, doi: 10.1016/j.cosrev.2017.09.001.

[18] AMADEOS, Supporting Facilities User Guide Rev. 0.1. (2016). Accessed: Jun. 1, 2021. [Online]. Available: https://blockly4sos.resiltech.com/user-guide.pdf

[19] “Amadeos supporting facility | SoS design using Blockly”. https://blockly4sos.resiltech.com/ (accessed Jun. 03, 2021).

[20] “The MITRE Corporation”. https://www.mitre.org/ (accessed May 24, 2021).

[21] “CWE - CWE-285: Improper Authorization (4.4)”. https://cwe.mitre.org/data/definitions/285.html (accessed May 24, 2021).

[22] “CVE - CVE-2021-28968”. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-28968 (accessed May 24, 2021).

[23] “CVE - Frequently Asked Questions”. https://cve.mitre.org/about/faqs.html#what_is_cve_id (accessed May 24, 2021).

[24] “National Institute of Standards and Technology,” NIST. https://www.nist.gov/ (accessed May 24, 2021).

[25] “NVD - Vulnerability Metrics”. https://nvd.nist.gov/vuln-metrics/cvss (accessed May 24, 2021).

[26] “Common Vulnerability Scoring System SIG,” FIRST — Forum of Incident Response and Security Teams. https://www.first.org/cvss (accessed May 24, 2021).

[27] “NVD - CVSS v2 Calculator”. https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator (accessed May 24, 2021).

[28] “NVD - CVSS v3 Calculator”. https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator (accessed May 24, 2021).

[29] “NVD - Detail - cpe:2.3:a:gnu:punbb:1.2.22:*:*:*:*:*:*:*”. https://nvd.nist.gov/products/cpe/detail/942847?keyword=cpe%3A2.3%3Aa%3Agnu%3Apunbb%3A1.2.22%3A*%3A*%3A*%3A*%3A*%3A*%3A*&status=FINAL,DEPRECATED&orderBy=CPEURI&namingFormat=2.3 (accessed May 24, 2021).

[30] “CAPEC - CAPEC-59: Session Credential Falsification through Prediction (Version 3.4)”. https://capec.mitre.org/data/definitions/59.html (accessed May 24, 2021).

[31] A. Brazhuk, “Semantic model of attacks and vulnerabilities based on CAPEC and CWE dictionaries,” International Journal of Open Information Technologies, vol. 7, pp. 38–41, 2019. [Online]. Available: https://www.semanticscholar.org/paper/Semantic-model-of-attacks-and-vulnerabilities-based-Brazhuk/105ad69ebff937ccaf41ae8392c844d0cb02a847

[32] K. Kanakogi et al., “Tracing CAPEC Attack Patterns from CVE Vulnerability Information using Natural Language Processing Technique,” in 54th Hawaii International Conference on System Sciences, 2021, p. 9. doi: 10.24251/HICSS.2021.841.

[33] C. M. Portela, M. Hoeve, F. H. Tan, and H. Slootweg., “Implementing an ISA/IEC-62443 and ISO/IEC-27001 OT Cyber Security Management System at Dutch DSO Enexis,” in 25th International Conference on Electricity Distribution, Madrid, Spain, 2019. [Online]. Available: https://www.cired-repository.org/handle/20.500.12455/404

[34] B. Jelacic, I. Lendak, S. Stoja, M. Stanojevic, and D. Rosic, “Security Risk Assessment-based Cloud Migration Methodology for Smart Grid OT Services,” ACTA POLYTECH HUNG, vol. 17, no. 5, pp. 113–134, 2020, doi: 10.12700/APH.17.5.2020.5.6.

[35] J. A. Wang, M. M. Guo, and J. Camargo, “An Ontological Approach to Computer System Security,” Information Security Journal: A Global Perspective, vol. 19, no. 2, pp. 61–73, Apr. 2010, doi: 10.1080/19393550903404902.

[36] Xiao H., Xing Z., Li X., Guo H., “Embedding and Predicting Software Security Entity Relationships: A Knowledge Graph Based Approach,” in International Conference on Neural Information Processing, Dec. 2019,pp. 50-63, doi: 10.1007/978-3-030-36718-3_5

[37] E. Hemberg et al., “Linking Threat Tactics, Techniques, and Patterns with Defensive Weaknesses, Vulnerabilities and Affected Platform Configurations for Cyber Hunting,” arXiv:2010.00533 [cs], Feb. 2021, Accessed: Jun. 21, 2021. [Online]. Available: http://arxiv.org/abs/2010.00533

[38] “MITRE ATT&CK®”. https://attack.mitre.org/ (accessed Jun. 21, 2021).

[39] V. N. L. Franqueira, T. T. Tun, Y. Yu, R. Wieringa, and B. Nuseibeh, “Risk and argument: A risk-based argumentation method for practical security,” in 2011 IEEE 19th International Requirements Engineering Conference, Aug. 2011, pp. 239–248, doi: 10.1109/RE.2011.6051659.

[40] C. Haley, R. Laney, J. Moffett and B. Nuseibeh, "Security Requirements Engineering: A Framework for Representation and Analysis," IEEE Transactions on Software Engineering, vol. 34, no. 1, pp. 133-153, Jan.-Feb. 2008, doi: 10.1109/TSE.2007.70754.

[41] L. Regainia, S. Salva, and C. Ecuhcurs, “A classification methodology for security patterns to help fix software weaknesses,” in 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Nov. 2016, pp. 1–8. doi: 10.1109/AICCSA.2016.7945693.

[42] S. Salva and L. Regainia, “A catalogue associating security patterns and attack steps to design secure applications,” Journal of Computer Security, vol. 27, pp. 49-74, Jan. 2019, doi: 10.3233/JCS-171063.

[43] B. Kordy, P. Kordy, S. Mauw, and P. Schweitzer, “ADTool: Security Analysis with Attack-Defense Trees,” in 2013 International Conference on Quantitative Evaluation of Systems, Jun. 2013, pp. 173-176, doi: 10.1007/978-3-642-40196-1_15

[44] “Talend Open Studio: Open-source ETL and free data integration,” Talend Real-Time Open Source Data Integration Software. https://www.talend.com/products/talend-open-studio/ (accessed May 24, 2021).

[45] “Talend Open Studio - Tutorialspoint”. https://www.tutorialspoint.com/talend/talend_talend_open_studio.htm (accessed May 24, 2021).

[46] “tMap • Talend Open Studio Components Reference Guide • Reader • Welcome to Talend Help Center”. https://help.talend.com/r/wDRBNUuxk629sNcI0dNYaA/mxzKD~8eLuNFSXH6LMi7qg (accessed May 24, 2021).

[47] “tXMLMap • Talend Open Studio Components Reference Guide • Reader • Welcome to Talend Help Center”. https://help.talend.com/r/wDRBNUuxk629sNcI0dNYaA/yZJFgdpcM_OEmIjL7ZnBng (accessed May 24, 2021).

[48] “ADTool”. https://satoss.uni.lu/members/piotr/adtool/ (accessed May 28, 2021).

[49] “KH Coder Index Page”. https://khcoder.net/en/ (accessed May 28, 2021).

[50] “CWE - CWE-13: ASP.NET Misconfiguration: Password in Configuration File (4.4)”. https://cwe.mitre.org/data/definitions/13.html (accessed Jun. 04, 2021).

[51] “CWE - CWE-12: ASP.NET Misconfiguration: Missing Custom Error Page (4.4)”. https://cwe.mitre.org/data/definitions/12.html (accessed Jun. 04, 2021).

[52] “CWE - CWE-664: Improper Control of a Resource Through its Lifetime (4.4)”. https://cwe.mitre.org/data/definitions/664.html (accessed Jun. 04, 2021).

[53] “CWE - CWE-1273: Device Unlock Credential Sharing (4.4)”. https://cwe.mitre.org/data/definitions/1273.html (accessed Jun. 04, 2021).

[54] “CWE - Chains and Composites”. https://cwe.mitre.org/data/reports/chains_and_composites.html (accessed May 24, 2021).

[55] “CAPEC - CAPEC-383: Harvesting Information via API Event Monitoring (Version 3.4)”. https://capec.mitre.org/data/definitions/383.html (accessed Jun. 04, 2021).

[56] “CAPEC - CAPEC-629: Unauthorized Use of Device Resources (Version 3.4)”. https://capec.mitre.org/data/definitions/629.html (accessed Jun. 04, 2021).

[57] “CAPEC - CAPEC-560: Use of Known Domain Credentials (Version 3.4)”. https://capec.mitre.org/data/definitions/560.html (accessed Jun. 04, 2021).

[58] “CAPEC - CAPEC-55: Rainbow Table Password Cracking (Version 3.4)”. https://capec.mitre.org/data/definitions/55.html (accessed Jun. 04, 2021).

[59] “CAPEC - CAPEC-70: Try Common or Default Usernames and Passwords (Version 3.4)”. https://capec.mitre.org/data/definitions/70.html (accessed Jun. 04, 2021).

[60] C. Dougherty, “Secure Design Patterns,” Software Engineering Institute, Technical Report ESC-TR-2009-010, 2009. Accessed: May 17, 2021. [Online]. Available: https://resources.sei.cmu.edu/asset_files/TechnicalReport/2009_005_001_15110.pdf

[61] D. M. Kienzle, M. C. Elder, D. Tyree, and J. Edwards-Hewitt, “Security Patterns Repository Version 1.0,” 2006. [Online]. Available: https://ecs.syr.edu/faculty/fawcett/handouts/CSE776/PatternPDFs/SecurityRepository.pdf

[62] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, and P. Sommerlad, Security Patterns: Integrating Security and Systems Engineering, 1st ed. Wiley, 2006, p. 600. Accessed: Jun. 24, 2021. [Online]. Available: https://www.wiley.com/en-ao/Security+Patterns%3A+Integrating+Security+and+Systems+Engineering-p-9780470858844

[63] K. Yskout, R. Scandariato, and W. Joosen, “Security Pattern Catalog,” 2015. [Online]. Available: https://people.cs.kuleuven.be/~koen.yskout/icse15/catalog.pdf

[64] “CWE - CWE-653: Insufficient Compartmentalization (4.4)”. https://cwe.mitre.org/data/definitions/653.html (accessed Jun. 04, 2021).

[65] D. Forsberg, “Privilege Separation: A Security Pattern,” in 4th Nordic Conference on Pattern Languages of Programs VikingPLoP, 2005, pp. 333-337. [Online]. Available: https://hillside.net/vikingplop/vikingplop2007/VikingPLoP_2003_2004_2005_Proceedings.pdf

[66] “CWE - CWE-521: Weak Password Requirements (4.4)”. https://cwe.mitre.org/data/definitions/521.html (accessed Jun. 04, 2021).

[67] E. Pedersen et al., “Net2DG Deliverable D1.2 – Initial Baseline Architecture,” Aug. 2018, Accessed: Jun. 1, 2021. [Online]. Available: http://www.net2dg.eu/1c254772_52ed_4f92_991a_9502725a16af.html

[68] N. Nostro et al., “Net2DG Deliverable D3.3 – ICT resilience mechanisms and verification,” Jun. 2020. Accessed: Jun. 1, 2021. [Online]. Available: http://www.net2dg.eu/1c254772_52ed_4f92_991a_9502725a16af.html

[69] “FAQ”. https://mqtt.org/faq/ (accessed May 30, 2021).

[70] M. Masse, “Introduction,” in REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces, O’Reilly Media, Inc., 2011, ch. 1, pp. 1–10. [Online]. Available: https://books.google.it/books?id=eABpzyTcJNIC&pg=PA1&hl=it&source=gbs_toc_r&cad=3#v=onepage&q&f=false

[71] “CWE - CWE-330: Use of Insufficiently Random Values (4.4)”. https://cwe.mitre.org/data/definitions/330.html (accessed Jun. 04, 2021).

[72] “CWE - CWE-1204: Generation of Weak Initialization Vector (IV) (4.4)”. https://cwe.mitre.org/data/definitions/1204.html (accessed Jun. 04, 2021).

[73] “CWE - CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection’) (4.4)”. https://cwe.mitre.org/data/definitions/89.html (accessed Jun. 04, 2021).

[74] “CWE - CWE-284: Improper Access Control (4.4)”. https://cwe.mitre.org/data/definitions/284.html (accessed Jun. 04, 2021).

[75] “CWE - CWE-307: Improper Restriction of Excessive Authentication Attempts (4.4)”. https://cwe.mitre.org/data/definitions/307.html (accessed Jun. 04, 2021).

[76] “CVE - CVE-2019-11779”. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-11779 (accessed Jun. 04, 2021).

[77] “CWE - CWE-131: Incorrect Calculation of Buffer Size (4.4)”. https://cwe.mitre.org/data/definitions/131.html (accessed Jun. 04, 2021).

[78] “CWE - CWE-1333: Inefficient Regular Expression Complexity (4.4)”. https://cwe.mitre.org/data/definitions/1333.html (accessed Jun. 04, 2021).

[79] “CVE - CVE-2018-20145”. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-20145 (accessed Jun. 04, 2021).

[80] “CWE - CWE-732: Incorrect Permission Assignment for Critical Resource (4.4)”. https://cwe.mitre.org/data/definitions/732.html (accessed Jun. 04, 2021).

[81] S. Salva and L. Regainia, “An Approach for Guiding Developers in the Choice of Security Solutions and in the Generation of Concrete Test Cases,” Software Quality Journal, vol. 27, pp. 675-701, Jun. 2019, doi: 10.1007/s11219-018-9438-2.