POLITECNICO DI TORINO

Master's Degree in Computer Engineering
Cybersecurity

Master's Degree Thesis

Cyber risk evaluation model for OT infrastructures

Supervisors

Prof. Cataldo Basile

Ing. Alessio Viticchié, PhD

Candidate

Marco Bor

July 2025

Summary

The work presented in this dissertation has been carried out in the context of an internal R&D project of the *Alphawaves Srl.* company, which hosted the activities of this thesis.

Specifically the project addresses the need for cybersecurity solutions within the world of operational technology (OT), by proposing a plug and play tool capable of performing automatic scanning, validation and reporting of the status of security within an OT network, in a continuous monitoring cycle.

Within the context of the project, this thesis focuses on the development of a Cyber risk evaluation model that receives as input a network description, containing information about host and network configuration, topology, and potential vulnerabilities; and generates possible scores to quantify the severity of threats on a scale from 0 to 10.

The design of the model is achieved by analysing relevant cybersecurity risk assessment frameworks that often treat risk factors in isolation, focusing solely on technical vulnerabilities, misconfigurations, or human errors, while neglecting their complex interplay.

The proposed model brings together diverse evaluation domains by merging quantitative risk metrics with qualitative factors into a cohesive and structured scoring system. This hybrid methodology enhances the accuracy and applicability of the risk evaluation process across various scenarios. Its methodology is based on the integration of four distinct components, each focused on assessing a specific aspect of risk:

- **Vulnerabilities**: essential for protecting corporate systems and preventing cyber attacks enabling proactive intervention.

- **Attack Graphs**: crucial because it allows for the identification and understanding of potential attack paths within a system.

- **Topology**: necessary for identifying critical points and improving infrastructure security, detecting potential bottlenecks and attack paths

- **Asset appraisal**: knowing its operational and economic impact allows for informed decisions on which measures to adopt to mitigate risks.

The validity of the risk evaluation model proposed in this work has been tested against a set of manually constructed OT network configurations, designed to replicate realistic industrial environments. In all cases, the model successfully computed individual threat scores for each host.

The thesis concludes with a discussion on practical implications, limitations, and future directions.

Acknowledgements

I express my sincere gratitude to AlphaWaves Srl. for giving me the opportunity to deepen my knowledge, particularly within the field of cybersecurity. Taking part in this project allowed me to grow both personally and professionally.

A particular thanks goes to my company supervisor, Ing. Alessio Viticchié. I also thank Prof. Alessandro Aliberti and Dr. Alberto Colletto for their continuous support during the thesis period.

I also express my heartfelt gratitude to their university supervisor, Prof. Cataldo Basile, whose guidance was instrumental throughout the development of this thesis. His support played a key role in shaping the final outcome and helped me approach complex topics with greater clarity and confidence.

To all of the colleagues at AlphaWaves, I extend my genuine appreciation for welcoming me into a dynamic and positive working environment and for making me feel like a valued member of the team.

I want to thank my friend Giulio Sunder as well for his friendship and constant support.

Finally, yet importantly, I am deeply grateful for my family. Their unwavering love and support throughout my academic journey and my life have been essential. Without them, this thesis would not have been possible.

Table of Contents

List of Tables

List of Figures

Acronyms

OT
operational technology

ICS
industrial control system

SCADA
supervisory control and data acquisition

HMI
human machine interface

PLC
programmable logic controller

RTU
remote terminal unit

DCS
distributed control system

EWS
engineering workstation

IED
intelligent electronic device

IIoT
industrial internet of things

AG

attack graph

CVSS

Common Vulnerability Scoring System

AV

Attack Vector

AC

Attack Complexity

CCSS

Common Configuration Scoring System

CMSS

Common Misuse Scoring System

EPSS

Exploit Prediction Scoring System

SCAP

Security Content Automation Protocol

PRISMA

Practices for Risk Indicator Scoring Maturity Assessment

CSF

Cybersecurity Framework

SSVC

Stakeholder-Specific Vulnerability Categorization

VAR

Value at Risk

Chapter 1

Introduction

In recent years, the exponential growth of digital technologies has profoundly transformed all sectors of industry and services, pushing companies to adopt increasingly interconnected and automated infrastructures. Among these, Operational Technology (OT), systems responsible for the monitoring and control of physical processes in sectors such as energy, manufacturing, transportation, and critical infrastructure, have undergone significant modernization. Once isolated and based on proprietary architectures, these systems are now often connected to corporate networks and the internet, following the paradigm of convergence with Information Technology (IT). This shift brings undeniable advantages in terms of flexibility, remote monitoring, and data analysis. However, it also introduces new and serious cybersecurity challenges.

Unlike IT environments, where cybersecurity strategies are mature and well-established, OT systems often lack adequate security measures. Many of them were designed decades ago with reliability and availability as their main goals, in contexts where the concept of cyber threats was virtually nonexistent. The resulting technological debt, coupled with the growing exposure to external networks, has made these infrastructures particularly vulnerable to attacks that can have devastating consequences—not only in economic terms but also in terms of safety, environmental damage, and public impact.

Despite the urgency of the problem, conducting a complete and automated security assessment of an OT infrastructure remains a complex and often manual process. Currently, most of the available tools are highly specialized, focusing on individual tasks such as vulnerability scanning, network topology discovery, or attack graph generation. These tools are largely designed with IT systems in mind, where assumptions such as dynamic reconfiguration, rapid patching, and cloud native architectures are valid. In contrast, OT environments feature tightly coupled hardware-software stacks, static configurations, and stringent uptime requirements, which make them less compatible with traditional IT-centric security solutions.

Additionally, commercial tools, though more integrated in some cases, tend to be opaque and closed-source, making them difficult to audit, customize, or adapt to specific industrial needs. These instruments often lack the transparency required to build trust in critical environments, and their proprietary nature can hinder their integration into existing operational workflows or research pipelines. Additionally, licensing costs and vendor lock-in represent further barriers for small and medium-sized enterprises. On the other side, academic research, while active and innovative, often results in proof-of-concept implementations or narrowly scoped theoretical frameworks that are not easily deployable in production-grade environments. Many proposed models remain in the experimental stage, with limited testing on real-world configurations, and without the necessary documentation or user interfaces for non-expert operators to utilize them effectively.

This thesis was developed within the research and development program of AlphaWaves Srl, with the goal of addressing this gap by designing and implementing a prototype reporting engine capable of automatically assessing the cybersecurity posture of OT networks. The motivation behind this work stems from the practical need to provide operators with a clear, structured, and actionable overview of system security, without requiring deep cybersecurity expertise or costly manual interventions. The long-term vision is to contribute to the creation of a plug-and-play device that, once connected to a network, can autonomously collect information, evaluate risks, and produce a detailed report highlighting vulnerabilities, critical assets, attack paths, and mitigation strategies.

The research objective of this thesis is to design and validate a risk evaluation model that, starting from structured input data concerning the network topology, known vulnerabilities, inter-device communication, and asset relevance, can generate a comprehensive risk score for each host.

The initial phase of the work consisted in a thorough review and comparison of existing risk assessment frameworks, including CVSS, EPSS, CCSS, CMSS, SSVC, and others. These frameworks, each with their own strengths and limitations, provided a valuable foundation for understanding how to quantify and prioritize cyber risks. However, none of them, alone, offered a solution suitable for the specific needs of OT environments. Therefore, the model proposed in this work draws inspiration from multiple frameworks and extends them by integrating contextual information from various layers of the system.

The risk assessment engine is made up of four specialized modules, vulnerability, attack graph, topology, and asset evaluation, each designed to analyze a different aspect of the security posture of the system. A key strength of the engine lies in its ability to unify diverse types of information under a common framework, where each module produces a normalized score on a uniform scale, allowing for direct comparison and final aggregation into a single risk score per host.

The vulnerability module evaluates the likelihood that a given vulnerability

will be exploited in practice. To do this, it integrates multiple data points such as scores quantifying the technical severity of a vulnerability and empirical data on how frequently the vulnerability has been used in known attack campaigns. This layered scoring approach allows the module to move beyond traditional static assessments and incorporate dynamic indicators from the real world, providing a more accurate estimation of the exploitation potential.

The attack graph module models how risk factors can be chained together by an attacker to achieve specific objectives. It receives as input an attack graph, where nodes correspond to hosts or system states and edges represent possible attack transitions. This structure enables the simulation of multistage attacks and captures the interdependencies among hosts. Furthermore, the module considers node attributes such as entry points or leaf nodes to better understand each host's strategic value in an attack chain. By quantifying metrics such as reachability, centrality, and branching factor, the module assesses each host's role within broader exploitation scenarios.

The topology module focuses on network structure and communication patterns. It takes as input a matrix that quantifies the volume and direction of packets exchanged between hosts. This data allows the module to infer how central or exposed a host is within the network. Hosts that exchange a high volume of traffic with many other devices are likely to be more critical for both operations and attack propagation. The matrix can be derived from real-time packet capture tools like PyShark or from prior scans, depending on the deployment context. This flexibility ensures that the evaluation remains adaptable to both static snapshots and evolving network conditions.

The asset appraisal module introduces the business and operational dimension into the analysis. Unlike the other modules, which rely on technical or traffic-based data, this module uses structured questionnaires to capture expert knowledge about the role and importance of each asset. The questions are designed to assess factors such as process criticality, downtime impact, and data sensitivity. Responses are processed through a weighted scoring mechanism, with some questions triggering conditional follow-ups to gather more nuanced information. In large-scale networks, the system also supports the grouping of similar assets to streamline the evaluation process while maintaining accuracy.

To validate the model, the engine was tested on three manually constructed OT network configurations, designed to simulate realistic scenarios with varying complexity and architectural choices.

Each network configuration used in this work has been carefully designed to reflect plausible real-world scenarios that could be encountered in OT infrastructures. These configurations incorporate a diverse set of components that are commonly found in industrial environments such as PLCs (Programmable Logic Controllers), HMIs (Human Machine Interfaces) for controlling the I/O field devices (sensors and

actuators); the SCADA server, the historian server, and EWS (Engineering Work Station), ensuring that the assessment performed by the risk evaluation engine is both relevant and representative of actual deployment conditions.

The results demonstrate the ability of the engine to capture and quantify relevant risk factors and to distinguish between more and less critical components even though, due to the scarcity of comparable methodologies and the limited availability of reference datasets or benchmarking studies in the current literature, it is difficult to quantitatively verify the model's accuracy. This limitation underlines the complexity of evaluating cybersecurity risk in diverse and evolving network architectures.

To address this shortcoming and reinforce the credibility and applicability of the proposed approach, a peer review process is being organized as a next step. The goal is to submit the methodology, results, and underlying assumptions to cybersecurity experts, industrial practitioners, and academic researchers. Their technical and domain-specific feedback will be instrumental in validating the framework from both a theoretical and practical standpoint.

Although the model is still in a prototypical stage, the results of this research are promising. The tool proves to be useful not only for producing technical assessments, but also for supporting strategic decisions, thanks to its ability to synthesize technical and business perspectives.

The remainder of the thesis is structured as follows:

Chapter 1 contains an introduction to the global project, and it briefly addresses the goals and the need for such a project.

Chapter 2 contains the thesis' background. It provides contextualization about the world of OT and describes how cybersecurity relates to this field.

Chapter 3 discusses the state of the art. It starts by giving a brief theoretical overview of the architectural design of the project. Then it proceeds to provide a detailed analysis of the existing literature that highlights the strengths and weaknesses of the existing evaluation framework.

Chapter 4 discusses the methodology that has been used to build the reporting engine. It describes the main steps such as how the input model was designed, how the different formulas work and how the different score generated are aggregated.

Chapter 5 details the experimental validation process and discusses the obtained results.

Finally, Chapter 6 presents the concluding remarks of the project, along with potential directions for future improvements and development.

Following, Appendix A has been included to provide the data structures referenced throughout the thesis, allowing readers to consult them without interrupting the narrative flow.

In conclusion, this thesis represents a first step toward the realization of an integrated and automated solution for OT security assessment. While much work

remains to be done to refine and extend the model, the foundations laid here offer a solid basis for future developments and potential adoption in real industrial contexts.

Chapter 2

Background

This chapter aims to offer the reader a broad introduction to the world of operational technology (OT), highlighting the critical importance of security within this domain. It also presents the key concepts and definitions that will serve as the basis for the discussions in the following chapters of the thesis.

This chapter is based on the definitions provided by the National Institute of Standards and Technology (NIST) in their guide to OT security [1].

2.1 OT Overview

As defined by the National Institute of Standards and Technology (NIST) [1], OT refers to a wide range of programmable systems and devices that interact directly with the physical environment or manage equipment that does so. These systems monitor and control physical processes, causing or detecting real-world changes.

OT systems consist of combinations of control components (e.g., electrical, mechanical, hydraulic, pneumatic) that act together to achieve an objective (e.g., manufacturing, transportation of matter or energy). The component of the system that focuses on generating the desired output is known as the process, while the component responsible for ensuring that the output adheres to predefined specifications is referred to as the controller.

A system can be configured in one of three ways: open loop, closed loop, or manual mode. In an open loop configuration, the output is regulated based on predefined parameters, without feedback influencing the control process. In contrast, a closed loop system incorporates feedback—where the output is monitored and reintroduced as input—allowing the system to adjust and maintain the desired control objective dynamically. Finally, in a manual mode configuration, the entire process is operated directly by human intervention, without automated control mechanisms.

OT differs significantly from traditional Information Technology (IT). OT is primarily concerned with the control and monitoring of physical systems, requiring real-time management to ensure both efficiency and safety. In contrast, IT focuses on data-centric functions—such as collecting, processing, and storing information—to support decision-making and communication.

The two domains also diverge in terms of security priorities. OT emphasizes availability, real-time performance, safety, and component longevity, as disruptions can directly impact physical processes. On the other hand, IT places greater importance on data integrity and confidentiality, often prioritizing these over performance speed.

2.1.1 Evolution of OT

OT has significantly evolved to keep pace with the rapid advancements in Information Technology (IT). Originally based on analog mechanical controls, OT systems have increasingly adopted digital embedded controls, enabling greater interconnectivity and integration with internet-based technologies. This transition has empowered edge devices with enhanced computing capabilities, allowing for decentralized and more efficient decision-making.

The influence of the Internet of Things (IoT) paradigm has been central to this shift, bringing advanced IT features—such as real-time data collection, analytics, automation, and enhanced connectivity—into OT environments. These innovations have improved productivity, operational safety, responsiveness, and system longevity across various industries.

Today, OT infrastructures are fundamental to the reliable operation of critical sectors such as energy, transportation, manufacturing, healthcare, and water management. Their role in monitoring and controlling physical processes makes them indispensable components of modern infrastructure.

However, the growing integration with IT also introduces new risks. As OT systems become more connected and capable, their exposure to cyber threats increases. The expansion of the attack surface, coupled with the rising complexity of smart devices, heightens the potential for vulnerabilities that could disrupt operations or compromise system integrity.

Addressing cybersecurity in OT thus requires dedicated strategies. Thorough risk and impact assessments must be carried out, and mitigation measures carefully tailored to the specific operational environment. Emerging technologies like edge computing and AI also offer new opportunities for real-time threat detection and increased resilience.

In conclusion, the evolution of OT represents both a remarkable leap in capability and a growing set of challenges. Safeguarding these systems demands an integrated approach—one that balances innovation with risk management and

fosters collaboration across sectors to ensure secure and sustainable operation.

2.1.2 OT System components

An OT system operates based on control loops that regulate physical processes through the coordinated interaction of sensors, controllers, and actuators.

Sensors collect real-time data from the physical environment, providing an accurate snapshot of the current process state. This information is sent to the controller, which serves as the system's decision-making core. The controller interprets sensor input and, using a control algorithm, calculates the appropriate control variables needed to guide the process toward the desired state. These calculations are typically influenced by predefined parameters known as set points.

Actuators are the physical components such as motors, valves, and switches—that execute the commands generated by the controller, directly affecting the behavior of the physical process.

For effective system management, human intervention is often necessary. Human-Machine Interfaces (HMIs) enable operators to monitor system performance, adjust set points, and reconfigure control logic based on evolving operational goals.

Additionally, a robust OT system must include diagnostics and maintenance capabilities to detect, prevent, and respond to faults or equipment degradation. These functions are essential for ensuring system reliability and longevity.

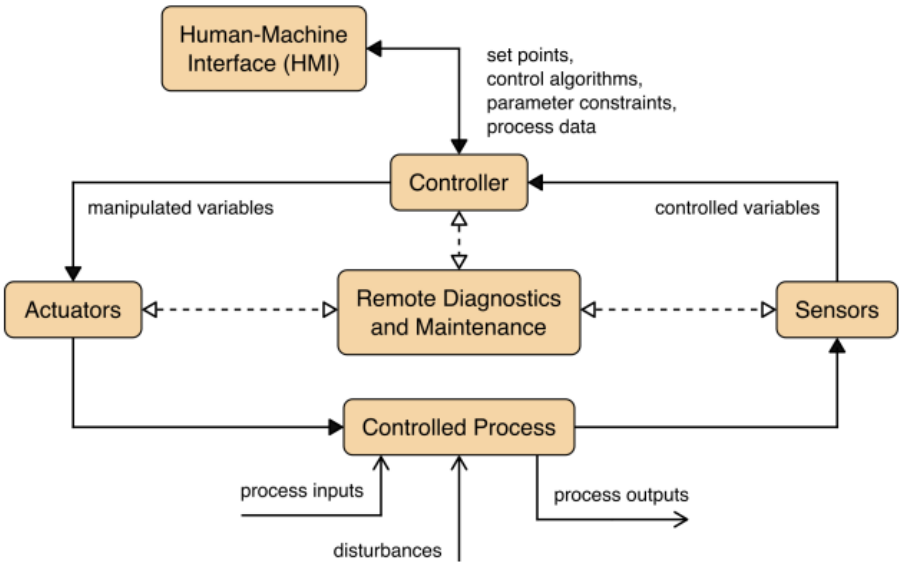

Figure 2.1: Standard OT system control loop taken from the NIST guide for OT [1]

A visual representation of the standard OT control loop is shown in Figure 2.1.

In terms of hardware architecture, an OT network typically includes a range of key components such as Human-Machine Interfaces (HMIs), Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs), Control Servers, Historian Servers, Engineering Workstations (EWS), I/O devices, and Intelligent Electronic Devices (IEDs). Each of these elements plays a specific role in the supervision, control, and automation of industrial processes.

PLC

As described in [2], programmable Logic Controllers (PLCs) serve as the primary controllers within an OT system. Essentially ruggedized industrial computers, PLCs are designed to operate reliably in harsh environments commonly found in industrial settings, such as extreme temperatures, electrical or mechanical interference, and inconsistent power quality.

Like conventional computers, PLCs comprise a central processing unit (CPU), memory, input/output (I/O) interfaces, and a communication module that enables network integration. Key features include the presence of a real-time operating system for time-sensitive processing, the ability to operate continuously with minimal maintenance, and the flexibility to be reprogrammed without the need for rewiring, as control logic is implemented in software.

To function, a PLC must be programmed with a control application that runs cyclically, reading sensor inputs via the I/O interface and issuing corresponding commands to actuators or other output devices. This enables real-time regulation of physical processes.

PLC programming is typically done using one or more of the five languages standardized by IEC 61131-3 [3]: Ladder Logic (LD), Function Block Diagram (FBD), Structured Text (ST), Instruction List (IL), and Sequential Function Chart (SFC).

Thanks to the integrated communication module, PLCs can be interconnected within a network. This allows them to exchange data with other systems for tasks such as centralized monitoring, remote visualization, and coordination with other controllers—facilitating distributed control and improved operational efficiency.

RTU

Like Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs) are used to monitor and control field devices within automated industrial processes. However, RTUs are generally more advanced and are specifically designed to handle larger, more complex tasks—often distributed across wide geographical areas.

[4] expalin the due to their suitability for remote and distributed operations, RTUs are typically integrated with Supervisory Control and Data Acquisition

(SCADA) systems (further discussed in Section 2.1.3). These systems enable centralized supervision and coordination of dispersed assets.

In contrast, PLCs are optimized for high-speed, localized control. They are particularly well-suited for centralized environments where fast and deterministic input/output processing is critical—such as in assembly lines, packaging equipment, or other time-sensitive industrial applications.

RTUs, by design, prioritize communication and adaptability over processing speed. They are equipped to interface with various control systems, often over wireless networks, to support remote operation in locations that may lack stable infrastructure or are physically separated.

Thanks to their lower complexity and cost, PLCs are generally preferred for automating simpler, centralized processes. They offer efficient performance, require less maintenance, consume less energy, and are easier to manage in stable environments. Conversely, RTUs are better suited for scenarios requiring remote monitoring, complex data exchange, flexible communication interfaces, and the execution of more sophisticated control logic.

In essence, the choice between PLCs and RTUs depends on the specific requirements of the application—PLCs excel in local, high-speed automation, while RTUs are ideal for managing distributed, remote, and complex control environments.

HMI

Human-Machine Interfaces (HMIs) serve as the critical bridge between human operators and industrial control systems. These interfaces provide real-time visualizations of process statuses and equipment conditions, enabling operators to monitor operations efficiently without the need for physical inspections across the facility. By centralizing data and presenting it through intuitive graphical displays—such as charts, graphs, and dashboards—HMIs streamline oversight and reduce the likelihood of human error.

Beyond passive monitoring, HMIs allow authorized personnel to interact with the system, facilitating tasks like adjusting operational parameters or responding to system alerts. This interactive capability not only enhances productivity but also contributes to improved safety and decision-making within industrial environments.

The integration of HMIs into industrial settings has transformed the way operators engage with machinery, shifting from manual, labor-intensive processes to more automated and user-friendly systems as described in [5]. This evolution supports the goals of modern industrial automation by promoting efficiency, accuracy, and responsiveness in process management.

EWS

An Engineering Workstation (EWS) is a computer within an OT network that provides authorized personnel with comprehensive access to system configuration and management tools, as described in [6]. While its role may appear similar to that of a Human-Machine Interface (HMI) in that both enable user interaction with operational processes the scope and capabilities of an EWS are significantly broader.

Unlike HMIs, which are typically limited by access controls and designed to offer only basic functionalities (such as monitoring process status or adjusting simple parameters), an EWS acts as a central hub for plant configuration and control. It generally provides a holistic view of the entire system, access to sensitive design and operational documentation, and software tools necessary to configure, program, and update critical control components like PLCs, RTUs, and servers.

Because of its extensive capabilities and access privileges, an EWS represents a high-value asset within an OT environment. Its compromise could have far-reaching impacts on plant operations. Although EWS and HMI systems may be targeted using similar attack vectors—due to their shared role as human interfaces within the network—the potential consequences of a successful attack on an EWS are typically much greater. This makes robust cybersecurity protections especially vital for these workstations.

Historian server

To support efficient process analysis and operational oversight, OT infrastructures commonly integrate specialized servers known as historians into their networks. These servers are designed to continuously collect, store, and organize live data from control devices—such as sensor readings, actuator outputs, and energy usage metrics—ensuring a comprehensive record of the system's behavior over time.

By maintaining a detailed historical log of process data, historian servers play a crucial role in enabling data-driven decision-making, as explained in [7]. The long-term storage of operational metrics allows for retrospective analysis, which can reveal inefficiencies, highlight recurring issues, and support strategic improvements aimed at reducing costs and maximizing productivity.

In addition to monitoring physical processes, historians also serve an essential function in system security. They record user interactions, alarms, and other key events, thereby enabling auditing and anomaly detection capabilities. This makes them valuable tools not only for operational performance but also for maintaining the integrity and safety of the OT environment.

Control server

A control server serves as a central component within an OT network, providing the supervisory control software that enables seamless coordination and communication across all lower-level control devices such as PLCs and RTUs. Acting as the operational backbone of the infrastructure, it plays a critical role in both the automation of physical processes and in ensuring system integrity.

Among its key functions is the collection and organization of real-time data from field devices via PLCs and RTUs. This includes metrics such as temperature, pressure, or actuator states, which are forwarded to the historian for long-term storage and analysis. By preserving this data, the control server supports anomaly detection, troubleshooting, and informed business planning based on historical performance.

Another essential responsibility of the control server is real-time monitoring and incident management. It continuously oversees system behavior, raises alarms when irregularities are detected, and dispatches notifications to alert personnel. This enables rapid response to potential threats or safety breaches, ensuring operational continuity and risk mitigation.

The control server also facilitates system visualization and reporting by aggregating operational data and presenting it through dashboards, charts, and diagrams. Automated generation of reports ensures that decision-makers and technicians maintain situational awareness across the plant's operations.

In terms of security and access control, the server enforces protection mechanisms such as user authentication, role-based access, encryption of communications, and activity logging. These measures help defend the system against internal misuse or external attacks.

Moreover, control servers often act as bridges between OT and IT by integrating with higher-level business systems, such as Manufacturing Execution Systems (MES) and Enterprise Resource Planning (ERP) platforms. Although these systems are IT-focused and will be discussed in more detail later, their interaction with control servers ensures the alignment of production operations with broader business objectives.

Finally, depending on the architectural design and the geographic distribution of the plant, multiple control servers may be deployed in a hierarchical structure, sharing responsibilities and ensuring redundancy for increased system resilience.

IED

The term IED, short for Intelligent Electronic Device, refers to advanced I/O components that combine control, monitoring, and communication capabilities within a single unit. Unlike traditional I/O devices that depend on controllers such as PLCs or RTUs to process data and manage interactions, IEDs are capable

of communicating directly with control servers, making them more autonomous within the OT network.

By embedding greater computational intelligence at the edge, IEDs contribute to increased system efficiency and reduced overall costs. Their ability to independently perform tasks such as data processing, decision-making, and system diagnostics reduces the load on centralized controllers and enhances the scalability of OT infrastructures.

This shift toward smarter, more capable edge devices reflects the broader influence of the Internet of Things (IoT) paradigm on industrial automation, where connectivity and decentralized intelligence have become key drivers of innovation and performance optimization.

2.1.3 OT network architecture

When integrating OT into a system, choosing the right architectural model is crucial and depends largely on the system's specific requirements and overall scope. A range of architectural options exists, each offering different strengths depending on the application context.

To guide this selection, several key factors must be evaluated:

- **Safety**: Concerns how effectively the system can detect and respond to hazardous situations. In many cases, human supervision remains critical to ensure operational safety.
- **Control timing requirements**: Encompasses time-sensitive aspects of automation, including speed, consistency, synchronization, and regularity—all essential for safe and reliable process execution.
- **Geographic distribution**: Refers to the physical spread of the system, which may range from a compact local installation to a wide-scale, geographically dispersed infrastructure like a national power grid.
- **Hierarchy**: Involves the organizational structure of systems operating across multiple locations and how control is aggregated or distributed.
- **Control complexity**: Relates to the sophistication required in automation and regulation to achieve specific operational objectives.
- **Availability**: Indicates required uptime levels, which in turn depend on the system's redundancy and fault tolerance.
- **Availability**: Considers the consequences of potential system faults and outlines how the architecture should mitigate or recover from such events based on their severity.

Taking these design factors into account, the following architectural models will be discussed in the upcoming sections: SCADA (Supervisory Control and Data Acquisition), DCS (Distributed Control Systems), and IIoT (Industrial Internet of Things). The discussion will place particular emphasis on SCADA, as it represents the architecture used in the referenced case study.

SCADA

SCADA, which stands for Supervisory Control and Data Acquisition, is a system architecture designed to monitor and control geographically distributed industrial processes from a centralized location. Its primary purpose is to gather operational data from field sites (typically managed by PLCs or RTUs) that locally oversee physical processes, and to present this data to human operators in an organized and accessible format. This allows personnel to supervise and manage the entire OT infrastructure efficiently and from a distance.

SCADA systems, as explained in [8], are particularly well-suited for large-scale geographically dispersed infrastructures such as oil and gas pipelines, water distribution networks, electrical grids, railway systems, and other forms of public transportation. In these scenarios, centralized oversight is essential: a failure in one remote station could potentially disrupt the operation of other interconnected components, making global visibility and coordination critical.

As illustrated in Figure 2.2, a typical SCADA control center is structured as a local area network (LAN) that includes a control server, human-machine interface (HMI), engineering workstation (EWS), historian, and gateway routers for communication with field stations.

The control server, described in detail in Section 2.1.2, acts as the system's central intelligence. It executes supervisory functions such as aggregating and logging data from field sites, initiating automated control actions based on sensor input, managing alarms and notifications, presenting data to operators through the HMI, and supporting system diagnostics, analysis, and reporting.

Due to the wide geographical scope of SCADA-managed systems, long-range communication technologies are essential. These may include wide area networks (WANs), satellite links, radio or cellular networks, and switched telephone or power line communication. These channels not only facilitate data exchange between field stations and the control center, but also enable remote access for diagnostics and maintenance, improving operational flexibility.

Each field site is equipped with sensors and actuators interfaced through PLCs or RTUs. These devices collect data and execute control actions under the guidance of the control center. In many cases, remote access features are also implemented to allow technicians to troubleshoot or update systems without requiring on-site intervention.

Additionally, some field sites may integrate Intelligent Electronic Devices (IEDs), which can communicate directly with the control server and perform embedded control tasks independently. In such scenarios, the use of a PLC or RTU becomes optional, as IEDs are capable of handling both communication and control functionalities on their own.

Given the critical nature of the systems managed by SCADA, redundancy and fault-tolerance are fundamental design principles. These ensure continuous availability, minimizing downtime and mitigating the consequences of system failures.

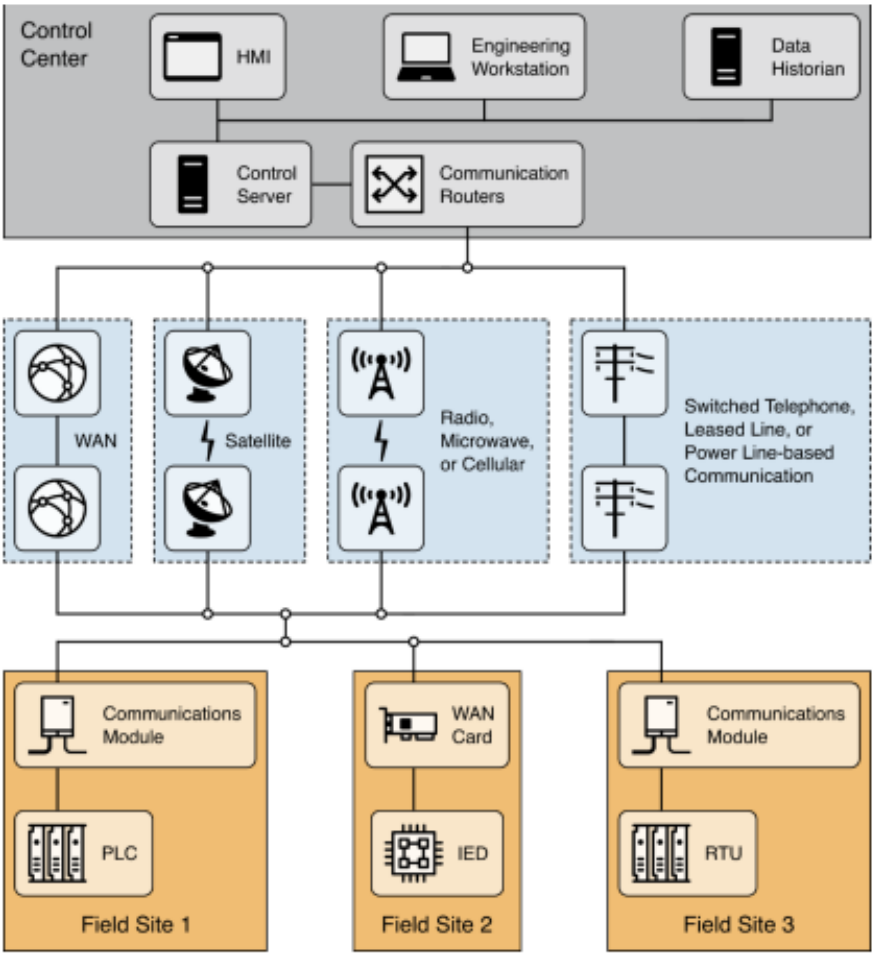

Figure 2.2: Standard SCADA architecture taken from the NIST guide for OT [1]

Depending on its size, the system could employ a number of control servers organized in a hierarchy, in order to ease the load on the central hub and increase

management efficiency by allowing the different nodes to work as independently as possible, as shown in Figure 2.3.

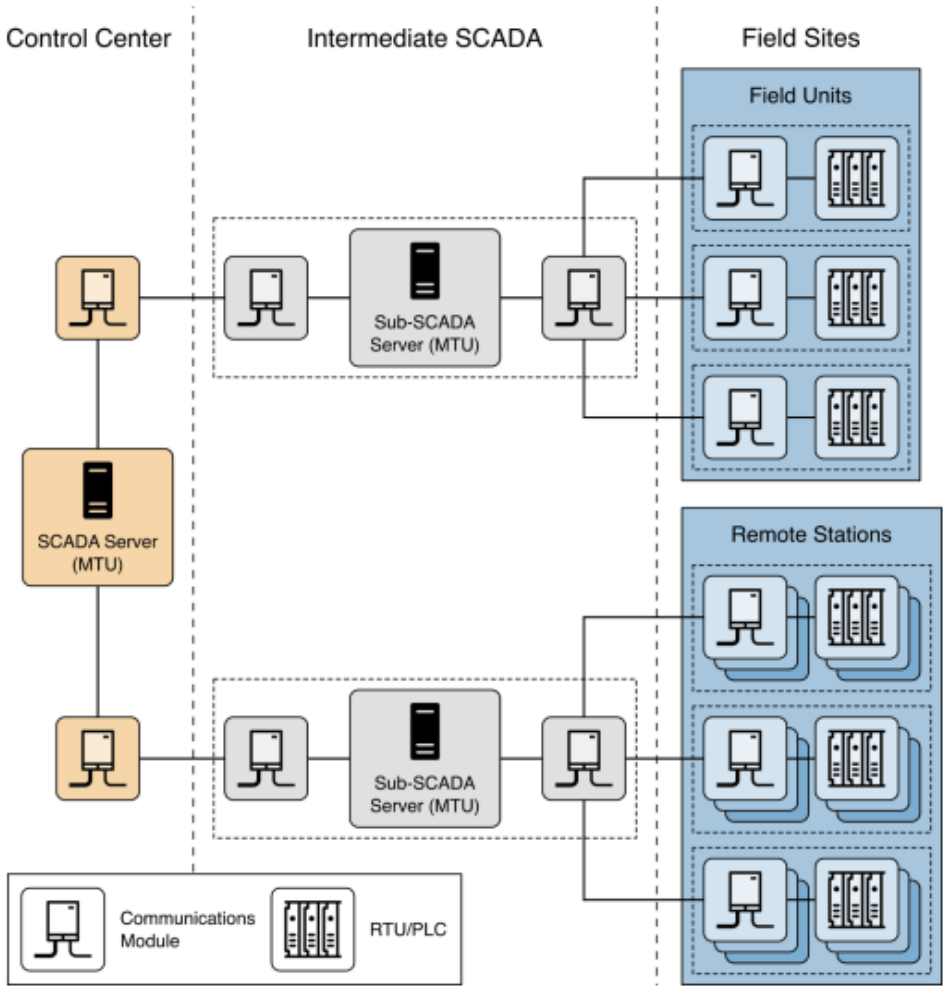

Figure 2.3: SCADA with multiple control servers taken from the NIST guide for OT [1]

DCS

Distributed Control System (DCS) is a control architecture commonly employed to manage production processes that are geographically confined to a single site or facility. Unlike SCADA systems, which are designed for wide-area supervisory control, DCSs are tailored for localized, high-reliability control.

Typical applications of DCSs include oil refineries, automotive manufacturing

plants, pharmaceutical production facilities, and similar industrial environments.

As illustrated in Figure 2.4, a DCS is composed of two primary layers:

a supervisory level, which includes components such as control servers, engineering workstations (EWSs), data historians, and operator consoles;

a field level, which comprises controllers (e.g., PLCs or dedicated control units), sensors, actuators, human-machine interfaces (HMIs), and other field devices.

The supervisory level oversees the distributed field stations by issuing setpoints and collecting operational data from the controllers. It typically hosts a data historian, responsible for archiving long-term process data, a control server, which directly communicates with field controllers to monitor and regulate their operations, an EWS, used by engineers and operators to configure, tune, and manage control logic and one or more operator consoles, which provide real-time process visualization and interaction.

The field level is divided into multiple field stations, each dedicated to controlling a specific part of the process. Each station is managed by a local controller — often a PLC, though more complex tasks may require specialized control units tailored to their function. In Figure 2.4, three field stations are shown: one managed by a generic PLC, the others by custom controllers.

Controllers execute either feedback or feedforward control loops, based on sensor inputs, to generate actuator commands that govern the underlying physical processes.

Devices at the field level such as sensors, actuators, HMIs, or remote terminals can be connected to their controller via a fieldbus rather than through individual point-to-point links. This bus-based communication reduces the need to route every signal through the controller, thereby improving efficiency. However, it requires support for specific industrial communication protocols, discussed in detail in Section 2.1.4.

Modern DCSs are typically integrated with the enterprise network to provide upper management with visibility into real-time production data. As shown in the top layer of Figure 2.4, this integration includes devices such as application servers, workstations, and printers, and links the DCS with systems like Management Information Systems (MIS) and Enterprise Resource Planning (ERP) platforms. These systems support decision-making processes through automation, data analytics, and real-time information flow between the production and business domains.

IIoT

As IT and OT systems increasingly converge, it becomes logical to adopt solutions traditionally used in one domain to benefit the other.

IIoT, or Industrial Internet of Things, brings the IoT paradigm from IT into the industrial world. The goal is to enhance industrial processes by integrating smart,

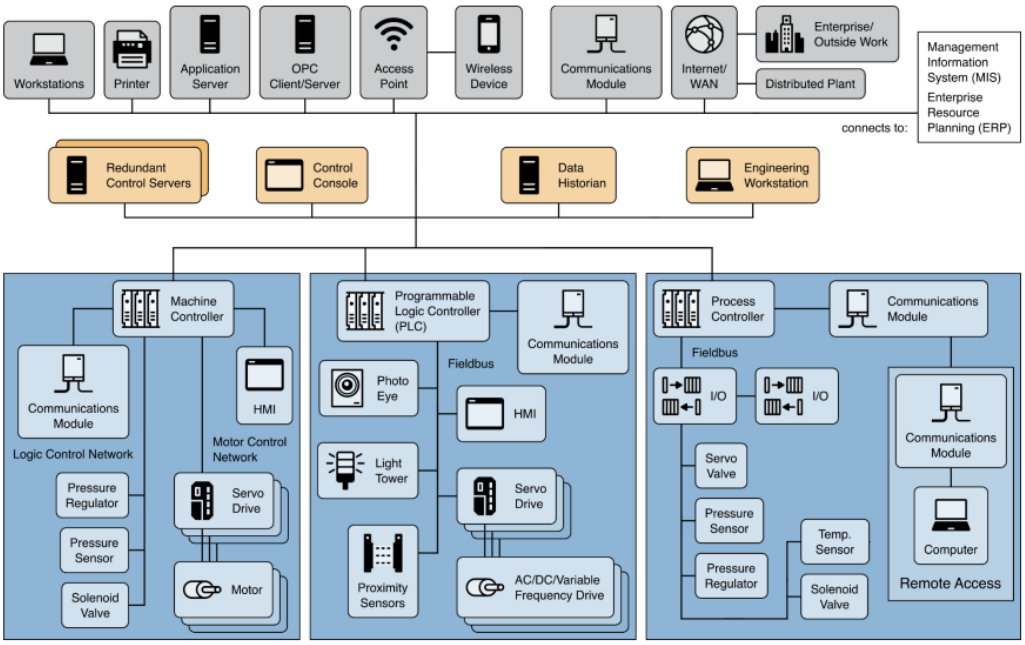

Figure 2.4: Standard DCS architecture example taken from the NIST guide for OT [1]

internet-connected devices into OT systems.

The typical IIoT architecture, shown in Figure 2.5, follows a three-tier model: edge tier, platform tier, and enterprise tier. This layered design organizes data processing across different levels — from raw data generated by field devices to business-level analytics — improving modularity, scalability, and efficiency. Cloud integration further enhances the system's data analysis capabilities.

The Enterprise Tier provides the user interface and supports business applications. It handles high-level decision-making by analyzing collected data and sending commands down to the other tiers.

Sitting between enterprise and edge, the platform tier provides general services like data processing, control command routing, and asset management. It does not focus on domain-specific logic but rather supports communication and coordination.

Connection to the enterprise tier is handled by a service network (e.g., a VPN), ensuring secure access to platform services by authorized enterprise applications.

The edge tier directly interfaces with field devices such as sensors and actuators over a proximity network. It collects real-time data, which can either be processed locally or sent upward for further analysis.

A key benefit of the edge layer is the ability to perform computation and control actions locally, reducing latency and boosting overall responsiveness by

decentralizing processing.

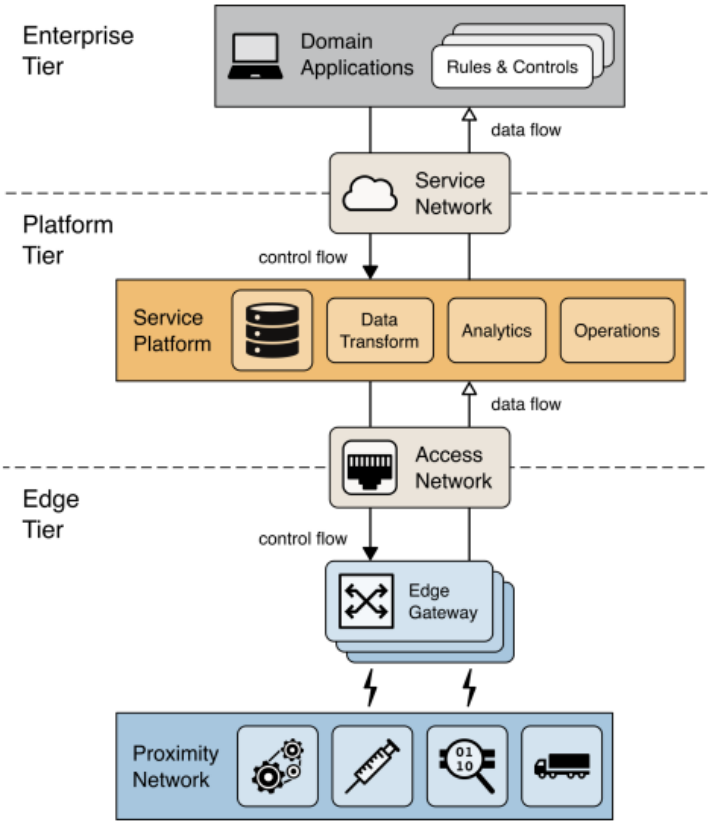

Figure 2.5: Standard IIoT architecture model taken from the NIST guide for OT [1]

2.1.4 Commonly used protocols

Over time, IT and OT have evolved with distinct networking standards, as they were originally developed for separate purposes with no expectation of convergence. As a result, OT networks rely on specialized communication protocols designed to meet the unique requirements of industrial environments—such as real-time control, reliability, and deterministic behavior.

These protocols differ significantly from those commonly used in IT systems. Among the most widely adopted in industrial automation are OPC-UA, MODBUS, and PROFIBUS/PROFINET.

OPC-UA

OPC-UA, which stands for Open Platform Communication Unified Architecture, is an open-source, platform-independent standard developed by the OPC Foundation for communication within industrial networks.

While a comprehensive overview can be found in [9], in brief, OPC-UA is a service-oriented, application-layer protocol widely used in OT environments. It enables both horizontal communication (e.g., between PLCs) and vertical communication (e.g., between PLCs and control or data aggregation servers).

OPC-UA supports both a client server communication model, based on either TCP/IP or HTTPS for the transportation of data, and a publisher subscriber communication model, compatible with protocols such as MQTT.

In typical SCADA systems, OPC-UA is often deployed in client-server mode, where PLCs act as OPC servers and control servers function as clients — requesting real-time field data or configuring process parameters.

Since its introduction in 2006, OPC-UA has seen multiple updates, including significant security enhancements such as support for authentication, data confidentiality, and integrity.

MODBUS

As stated in [10], MODBUS is a legacy protocol developed by Modicon in the 1970s to enable point-to-point communication between their PLCs. Due to its simplicity and widespread adoption, it was eventually made openly available and has since become a standard communication protocol for PLCs in the industrial automation landscape.

Like OPC-UA, MODBUS is commonly used for interfacing PLCs with supervisory systems, but it differs significantly in structure and capabilities. While OPC-UA is an application-layer protocol that operates primarily in a client-server model and supports different transport protocols, MODBUS offers several variants depending on the physical and network layers involved:

- Modbus RTU operates over serial connections, using a compact binary format and basic error checking, in a master-slave configuration.
- Modbus TCP/IP leverages the TCP/IP stack for higher-speed and long-distance communication, although this comes at the cost of added overhead and complexity.

From a security perspective, MODBUS lacks many of the protections found in more modern protocols like OPC-UA. In particular, being a much older protocol, MODBUS does not offer authentication or protection against unauthorized commands, making it more vulnerable in unsecured environments.

PROFIBUS/PROFINET

As explained in [11], PROFIBUS is a serial bus communication protocol widely used in OT systems, primarily for field-level interactions between PLC controllers and connected field devices such as sensors and actuators arranged in a bus topology.

Depending on the system's needs, there are two main PROFIBUS variants to choose from: Profibus DP and Profibus PA. Profibus DP (Decentralized Peripherals) is optimized for high-speed communication, making it suitable when fast data exchange is critical. In contrast, Profibus PA (Process Automation) prioritizes safe operation in hazardous environments, trading off transmission speed to ensure reliability and safety.

In comparison, PROFINET is an Ethernet-based protocol that utilizes the benefits of Ethernet and TCP/IP standards. Facilitates the seamless integration of factory and process automation systems, supports higher data rates, and can handle more complex network configurations, making it suitable for modern industrial environments.

2.2 Security in OT

For decades, cybersecurity has not received the level of attention it deserves within the broader field of computer science. However, this trend has shifted in recent years, with cybersecurity gaining widespread recognition for its critical importance.

In particular, the past five years have underscored just how vital technology and internet connectivity have become to the smooth functioning of daily life in domains such as healthcare, social interactions, and the workplace. These developments have also revealed the significant gaps that still exist in cybersecurity in various sectors.

One of the most affected and vulnerable areas is undoubtedly OT. This is particularly alarming considering OT's role in managing essential and often critical infrastructure. If a cybercriminal were to compromise such a system, the consequences could be catastrophic.

The root of many security weaknesses in OT systems lies in the fact that, historically, they were kept isolated from external networks, a practice known as maintaining an *air gap*. For years, this approach served as the main, and sometimes sole, security measure against cyber threats. While the air gap significantly reduces exposure to external attacks, it does little to address insider threats, which now account for over 60% of reported security incidents worldwide.

As OT systems are increasingly connected to the internet to support remote operations, diagnostics, and data analytics, their reliance on the *air gap* for protection becomes obsolete. This transition exposes previously contained vulnerabilities to the open internet, drastically expanding the attack surface.

This scenario illustrates a failure to apply one of the fundamental principles of cybersecurity: defense in depth. This principle advocates for a layered security approach, where multiple defensive mechanisms are in place so that, should one layer fail, others remain to protect the system.

2.2.1 Security priorities in OT versus IT systems

Although OT and IT systems are increasingly integrated, they still exhibit distinct characteristics due to the fundamental differences in their purposes.

OT is specifically designed to manage and control physical processes, whereas Information Technology (IT) primarily supports data-driven functionalities such as communication, computation, and analysis.

Because the priorities of OT and IT differ, their cybersecurity approaches must also be distinct, tailored to the unique requirements and constraints of each environment.

As stated in [1] the main properties around which security must be built for OT scenarios are:

- **Timeliness and performance requirements**: typically OT systems cannot handle time delay and jitter, as most of the processes they control are time critical and therefore require real-time responsiveness in order to enforce deterministic behaviour. From a cybersecurity standpoint this requirement might rule out the application of solutions that could introduce some sort of overhead, such as traffic encryption.

- **Availability requirements**: availability is another important characteristic of OT systems, as they are responsible for running important processes that cannot afford downtime and must therefore be available at all times. This normally entails different forms of redundancy to maintain operation in case of components becoming unavailable. Due to this requirement, cybersecurity solutions must be exhaustively tested before deployment, as it is not possible to interrupt the system's operation every time a security update needs to take place, or when an accident happens, on the contrary to what happens in IT. It is although still possible to perform security updates during the lifetime of the system, but they must happen during previously planned periods when it is alright to stop the system's operation, therefore they cannot be scheduled frequently.

- **Risk management requirements**: arguably the most important concern of OT is safety as opposed to IT which is more concerned with maintaining confidentiality and integrity of data. Cybersecurity solutions must therefore focus on maintaining safety at all times by implementing measures such as

proper access control, least privilege, adequate network segmentation, safety shutdown mechanisms, alarm notification system.

- **Communications**: even though nowadays OT systems are becoming more compatible with common IT protocols such as ethernet/IP, they still rely on different ad-hoc protocols some of which could even be proprietary. These type of protocols did not get the same focus on security improvement that their IT counterparts did, therefore they could present several potentially exploitable vulnerabilities.
- **Managed support**: most modern day IT systems make use of third party software to provide different functionalities (e.g. online payment processing). In OT however, many systems are supported by a single vendor, not allowing for any third party solutions to be used without breaking prior licenses and service agreements.
- **Component lifetime**: in contrast to IT technology, in which components typically last for short periods of time (about 4-5 years) due to the rapid evolution characterizing this branch of technology, components of an OT system are built to last for longer time spans (up to 15 years) allowing for prolonged operation before having to block any controlled processes for hardware upgrades. This point also plays a part in security, as it requires a lot of care and attention in selecting good hardware equipment and ensuring they meet the necessary security standards before including them in the operation cycle.
- **Component location**: as opposed to IT components, it is common to find OT components hosted in remote facilities, which are difficult to reach physically by human personnel due to the nature of the processes they control. Therefore security measures must also address this characteristic by providing alternative ways to react or reach these devices in a secure way in case something goes wrong.

2.2.2 Security assessment process

The process of security validation is the step by step procedure that is carried out to frame the overall security status of a system, in order to notify the people of interest about the discovered issues and potential consequences that could happen if not promptly handled.

This procedure is extremely important when it comes to securing any type of system. If not performed accurately it could lead to the exposure of the system to all sorts of threats, some of which could be critical, therefore major care must be taken when executing this process.

The main steps that constitute the validation process are:

- **Reconnaissance**: regards the process of gathering all of the information about the system (hosts, subnets, protocols, services, vulnerabilities, topology, asset values, and other useful data) that is needed to conduct the following steps. This could be achieved for example through passive/active scanning techniques by making use of popular open source scanning tools such as Nmap or Nessus. This is the most important part of the process, as all of the following steps depend on the information gathered during the this phase, therefore if the produced data weren't accurate enough many attacks could go undetected, causing a cascading effect in the rest of the procedure.

- **Attack modelling and planning**: this point is about constructing a model for representing and enumerating the potential attacks that could be attempted by malicious actors to compromise the system by exploiting discovered vulnerabilities and network configuration.

- **Attack validation**. concerns selecting and executing the appropriate tasks to test the actual exploitability of the potential attack paths, modelled at the previous step, as some may be false positives.

- **Attack risk analysis**: involves the definition of metrics for expressing the risk associated with the discovered potential attacks in a measurable way. The overall risk should take into account metrics describing probabilities of threat occurrences, asset values, impact in case of failures.

- **Mitigation strategy support**: indicates how the system should be modified in order to reduce the risk parameters, calculated at the previous step, so that they comply with defined acceptable value ranges. The proposed modification strategies could be of different forms, such as software vulnerability patching, introduction of firewalls for providing access control and network segmentation, addition of servers for increasing redundancy and ensuring greater availability, termination of all non necessary open ports, and so on.

- **Reporting**: revolves around the production of a human readable report detailing all the information generated in the previous steps.

In order to automate this process, suitable models should be defined for structuring all of the required data so that it can be represented in a standard way, allowing it to be understood and processed automatically by the various modules responsible for managing the previously listed steps.

2.2.3 Risk modeling

Risk modeling plays a crucial role in the broader domain of cybersecurity, especially in contexts where the protection of complex and interconnected systems, such as OT environments, is essential. This discipline focuses on the identification, quantification, and evaluation of risks that may compromise the confidentiality, integrity, and availability of digital and physical assets. The goal of risk modeling is to provide decision-makers with a structured and data-informed understanding of potential threats and their consequences, allowing them to design and implement effective mitigation strategies.

As stated in [12], in recent years, as IT and OT systems have progressively converged, the complexity and exposure of industrial networks to cyber threats have dramatically increased. This has emphasized the necessity of adopting formal risk modeling frameworks that can account not only for technical vulnerabilities but also for systemic interdependencies, human factors, and business impact.

Commonly used tools in this domain include attack trees, fault trees and Monte Carlo simulations. These models are often embedded within larger risk management frameworks and are used to simulate possible attack paths, calculate the likelihood of various threat scenarios, and estimate the potential impact in economic or operational terms.

A key principle in risk modeling is the identification of threats, vulnerabilities, assets, and controls. By systematically analyzing how a threat actor might exploit a vulnerability to compromise an asset, and evaluating the effectiveness of existing controls, analysts can estimate the residual risk and recommend additional countermeasures.

Risk modeling is not a static process, it requires continuous updates and revisions to reflect changes in the threat landscape, technological advancements, and the evolving operational context. This dynamic aspect is particularly relevant in critical infrastructure and industrial settings, where even minor disruptions can lead to significant safety, financial, or environmental consequences.

Integrating robust risk modeling practices into the cybersecurity lifecycle is essential for organizations seeking to anticipate potential incidents, justify security investments, comply with regulatory standards, and increase their overall cyber resilience.

2.2.4 Risk modeling in other fields

While risk modeling plays a vital role in the assessment and mitigation of cybersecurity threats its application is not limited to the technological domain. One of the most mature and long-established sectors where risk modeling has been extensively used is the insurance industry. In this context, risk models serve as the foundation for virtually all core business decisions, from underwriting policies to determining

premiums, reserving capital, and forecasting potential losses. These models analyze vast amounts of data, including demographic variables, historical loss data, behavioral trends, macroeconomic indicators, and even geographic information, to estimate the probability and financial impact of specific risks, such as natural disasters, car accidents, or health-related claims.

Due to the strategic and economic value that accurate risk prediction provides, most insurance companies invest heavily in the development and refinement of proprietary modeling techniques. These models often leverage advanced mathematical frameworks, statistical inference, and, increasingly, machine learning algorithms to improve prediction accuracy and operational efficiency. Given their importance, companies are generally reluctant to disclose the inner workings of these models. Unlike the more transparent models developed within academic research or regulatory contexts, risk models used in the insurance sector are typically kept confidential and treated as valuable intellectual property. This secrecy is maintained to preserve a competitive advantage and to avoid disclosing sensitive assumptions or limitations that might be exploited by competitors or lead to regulatory scrutiny.

As a result, much of the risk modeling in the insurance domain functions as a "black box", where only the inputs and outputs are visible, while the internal logic remains inaccessible to the public or even to many internal stakeholders. Despite this, the mathematical foundations and methodological approaches developed within the insurance industry have influenced risk modeling practices in other fields, particularly in areas like operational risk management, financial forecasting, and cybersecurity. In fact, many of the probabilistic and statistical techniques originally devised for actuarial science are now being adapted to model cyber risk, predict system failures, and support decision-making in complex technical environments.

2.2.5 Problem statement

As discussed in Section 2.2.3, one of the main challenges in developing an effective risk assessment model, particularly in complex systems such as those found in industrial or cyber-physical domains, is the difficulty of moving beyond isolated evaluations of individual risk components to adopt a general view of the system as a whole.

Traditional approaches to risk analysis often focus on specific assets, threats, or vulnerabilities in a fragmented way, treating them as independent variables. While this method can yield useful insights into particular weak points, it tends to overlook the broader interdependencies, cascading effects, and systemic behaviors that can arise when multiple elements interact. In reality, risk is not simply the sum of individual exposures, but the result of a network of dynamic relationships between technology, processes, people, and external factors.

This interconnectedness means that the impact of a seemingly minor vulnerability

in one part of the system could propagate and cause disproportionate damage elsewhere. Therefore, designing a risk model that integrates all these dimensions into a unified framework considering technical and contextual aspects simultaneously is a complex but necessary task. Without this systemic perspective, risk assessments may remain incomplete, potentially leading to inaccurate prioritization of resources and a false sense of security.

The following chapters will address the steps that this work took, in the scope of the company project mentioned in the introduction, to get a step closer to realising a risk model for OT infrastructures. And in doing so, it will also analyse relevant works available in the literature.

Chapter 3

Literature Review

Even today, a significant amount of human effort is required for conducting cybersecurity assessments of networked infrastructures. The inherent complexity of such tasks makes it difficult to develop fully automated solutions that can replace human expertise. This manual approach is not only time-consuming and resource-intensive, but also prone to errors, as it relies heavily on the analyst's technical knowledge and visual inspection.

Over time, a variety of tools have been developed to partially automate specific stages of the security assessment process such as vulnerability scanners, exploit frameworks, and auxiliary modules. While these tools were primarily designed for IT environments, they have become increasingly relevant in OT contexts as the boundary between IT and OT continues to blur. Despite the advantages these tools bring in terms of efficiency and coverage, there is still no standardized, comprehensive solution that can perform a full cybersecurity assessment with minimal human intervention, particularly for OT systems.

This gap highlights the pressing need for an integrated, end-to-end tool capable of automatically analyzing OT infrastructures in their entirety. Such a solution would not only streamline the assessment process but also significantly reduce the associated costs and reliance on highly specialized personnel, offering a more scalable and error-resilient approach to cybersecurity in critical environments.

3.1 Context of the project and scope of this work

The global context, within which the project collocates itself, is the creation of a standard solution for automating the process of security validation of OT infrastructures, therefore addressing the problem highlighted in the introduction of this chapter.

In order to achieve this, the proposed tool must be capable of performing the

main security assessment steps described in Section 2.2.2. All of which should happen in a continuous update fashion, allowing to adapt when changes take place.

From a physical stand point the proposed tool is a plug-and-play device that once connected to a network should be able to dynamically assess its security without the need for prior knowledge (black box approach). However, if more detailed results wish to be produced, it could be possible for administrators to provide the tool with some additional information about the network (grey box approach).

The general workflow of the project is given in Figure 3.1.

As can be seen in the figure, the proposed tool relies on five engines in order to carry out a complete security assessment process:

- **Reconnaissance engine**: is the engine responsible for scanning the network and producing a detailed description containing information such as device configurations, network configuration, network topology, exposed vulnerabilities, and access restrictions.
- **Formal verification engine**: is the engine responsible for computing possible attack paths and generating a validation task execution plan based on the information gathered by the reconnaissance engine.
- **Task execution engine**: is the engine responsible for running the validation tasks selected by the formal verification engine, in order to detect which discovered attack paths are actually exploitable and which are false positives. The output of this step is the set of the refined attack paths obtained by eliminating the false positives.
- **Risk assessment engine**: this engine is responsible for ranking the attack paths, obtained by the combined effort of the previous engines, based on risk analysis metrics and formulas.
- **Reporting engine**: is the engine responsible for summarizing all the useful information generated during the whole process and use it to automatically generate a complete report about the overall status of security within the system under evaluation.

With respect to this architecture, this work focuses on the design and development of a prototype for the reporting engine.

3.2 Need for evaluation framework

Conducting comprehensive cyber risk reporting remains a complex task for organizations due to several inherent challenges. A primary difficulty is the scarcity and

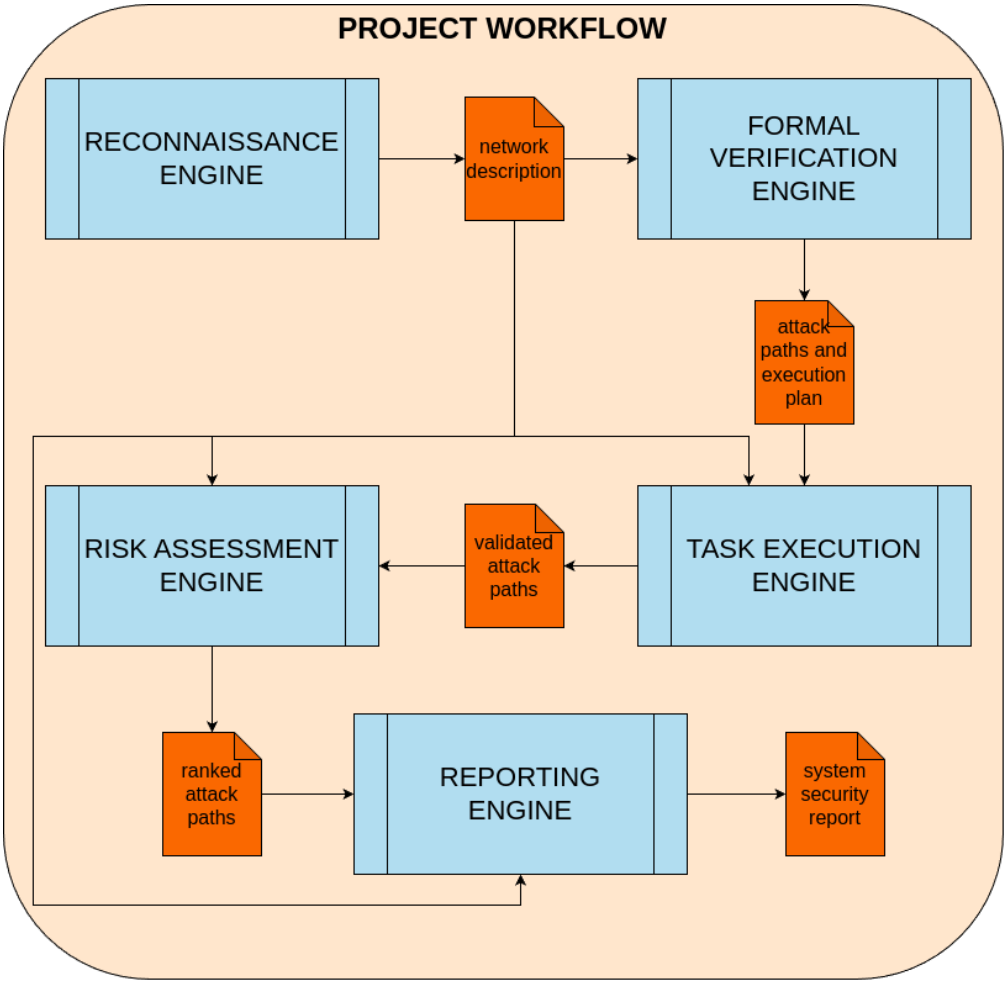

Figure 3.1: Workflow of the full company project

inconsistency of reliable data on cyber incidents. As highlighted in [13], the lack of accessible and standardized data hampers the accurate identification, assessment, and valuation of cyber risks, making it challenging to estimate the potential consequences of cyber incidents effectively.

Furthermore, the dynamic and evolving nature of cyber threats complicates the establishment of consistent reporting metrics and benchmarks.

In order to carry out a meaningful reporting activity within the context of risk assessment, it is essential to have a solid understanding of the existing evaluation frameworks.

These frameworks provide the foundational methodologies, metrics, and structures needed to interpret, and organize risk related data in a coherent and standardized way. Without proper knowledge of these models, reporting risks becoming

inconsistent or not aligned with industry best practices.

Analyzing the available frameworks not only helps in selecting the most suitable one for the specific context, but also ensures that the reporting output is both actionable and compliant with regulatory and organizational requirements.

3.3 Comparative Overview

In the following section, a selection of the most relevant cybersecurity risk assessment frameworks will be presented and briefly described, with a focus on their structure and applicability, analyzing their strengths and weaknesses.

This overview aims to provide a foundation for comparison and to support the development of a more integrated and context aware risk evaluation model.

3.3.1 Common Vulnerability Scoring System (CVSS)

The Common Vulnerability Scoring System (CVSS) is a structured and widely adopted framework designed to quantify the severity of software vulnerabilities on a scale from 0 to 10. At its core, CVSS is organized into three distinct metric groups: Base, Temporal, and Environmental, which respectively reflect the intrinsic nature of the vulnerability, its changing characteristics over time, and its relevance to a specific user environment.

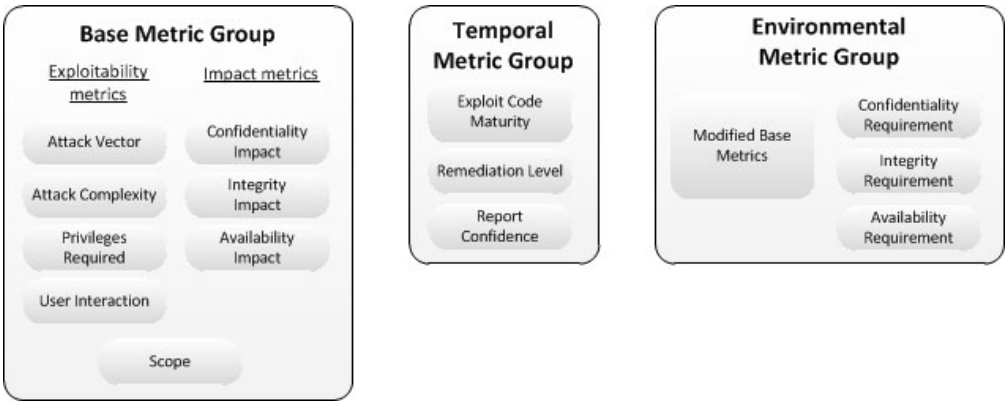

Figure 3.2: CVSS v3.0 Metric Groups

As described in [14] the Base Score is the cornerstone of ther framework, representing the intrinsic severity of a vulnerability that is constant over time and independent of external factors like exploit availability or organizational context. It is derived from two sub-scores: the Exploitability Score and the Impact Score, which together form a numerical value ranging from 0.0 to 10.0. The Exploitability

Score quantifies how easily the vulnerability can be exploited and is composed of four key metrics:

- **Attack Vector (AV)**: describes how an attacker can exploit the vulnerability. It ranges from Network (most remote and severe) to Physical (least severe), with values like Adjacent Network and Local in between.
- **Attack Complexity (AC)**: indicates the conditions beyond the attacker's control that must exist for the exploit to succeed. A Low value implies few or no conditions, while High suggests significant prerequisites.
- **Privileges Required**: specifies the level of privileges an attacker must possess before successfully exploiting the vulnerability. Fewer privileges result in a higher severity.
- **User Interaction**: denotes whether user interaction is required for the attack to succeed. Exploits requiring no interaction are more critical.

The Scope metric affects both the Exploitability and Impact calculations. It determines whether a vulnerability in one software component can impact resources beyond its own security authority. If the scope is Changed, it means the vulnerability can affect other components, increasing the severity.

The Impact Score evaluates the consequences of a successful exploit on three core security properties:

- **Confidentiality Impact**: measures the extent to which sensitive information is exposed.
- **Integrity Impact**: assesses the degree to which data can be modified or tampered with.
- **Availability Impact**: evaluates the potential disruption to the availability of the system or service.

Each impact metric is rated as None, Low, or High, and the combination of these ratings determines the overall Impact Score. The final Base Score is calculated using a predefined formula standardized by the CVSS specification, which considers both the Exploitability and Impact Scores, as well as the Scope.

The Temporal Score adjusts the Base Score to reflect the current state of exploitability, remediation, and the confidence in the vulnerability report. It includes three metrics:

- **Exploit Code Maturity**: indicates the availability and sophistication of exploit tools.

- **Remediation Level**: reflects the extent of remediation available (such as official fixes or workarounds).
- **Report Confidence**: gauges the credibility and detail level of the vulnerability report.

These metrics allow the Temporal Score to evolve over time, aligning with changes in threat landscape or vendor response.

The Environmental Score allows tailoring of the severity rating to the specific context of a particular organization or environment. It includes modified versions of the Base metrics to account for differences in deployment conditions or compensating controls. These adjustments allow the Environmental Score to better represent the actual risk posed by a vulnerability within a specific operational setting.

When used together, the Base, Temporal, and Environmental Scores form a comprehensive scoring system that supports both broad vulnerability comparison and precise, context-sensitive risk prioritization.

Strengths and weaknesses

In [15] we can read that CVSS offers several notable strengths that have contributed to its widespread adoption in all industries. One of its main advantages is standardization, as it provides a consistent and structured way to assess and compare vulnerabilities across different systems and organizations. Its modular structure enables both general-purpose and context-specific assessments, making it flexible for various use cases. Furthermore, CVSS is easy to understand for both technical and non-technical stakeholders, thanks to its intuitive scoring scale from 0 to 10 and clear metric definitions.

However, despite these strengths, it also presents some significant weaknesses. One key limitation is its focus on technical severity rather than actual risk, in fact, it does not directly account for exploitability in the wild, system interdependencies, or threat actor behavior. Additionally, the Environmental and Temporal metrics are often underutilized in practice, leading to overreliance on the Base Score, which may result in misleading prioritization. Studies such as [16] have highlighted how CVSS scores do not always correlate well with real-world exploitation patterns, questioning its effectiveness as a stand-alone risk prioritization tool. As a result, while CVSS remains a valuable component of vulnerability management, it is best used in combination with other risk assessment methods and threat intelligence sources.

3.3.2 Common Configuration Scoring System (CCSS)

The Common Configuration Scoring System (CCSS), introduced by NIST in 2010, is a specialized framework designed to evaluate the severity of software security

misconfigurations, complementing CVSS by focusing specifically on configuration related vulnerabilities. Structured similarly to CVSS, CCSS, as stated in [17], uses a combination of Base, Temporal, and Environmental metrics to derive a numerical severity score between 0 and 10 for each configuration issue.

The Base metrics assess the intrinsic properties of a misconfiguration, such as whether it can be actively exploited or only passively exploited, and its complexity and scope. Temporal metrics reflect how exploit availability and vendor remediation evolve over time, and Environmental metrics adjust scores based on context, like the organizational impact of a misconfiguration or the proportion of systems affected.

One of its key applications lies in automated configuration auditing, where it helps prioritize remediation efforts by highlighting the most critical misconfigurations in a system. CCSS scores can also be integrated into risk management frameworks, aiding security teams in aligning configuration management with broader cybersecurity goals. It serves as a valuable tool in compliance reporting, helping organizations demonstrate adherence to best practices and regulatory standards. CCSS allows organizations to perform context-aware risk assessment, enabling more informed decisions regarding mitigation strategies and resource allocation.

Strengths and weaknesses

The CCSS offers several strengths that make it a valuable tool in assessing the security impact of system misconfigurations. One of its key advantages is its structured and standardized approach to scoring, which promotes consistency and comparability between different environments. By incorporating different metrics, CCSS allows for a more comprehensive and context-sensitive evaluation than simpler scoring methods. This enables organizations to tailor the severity assessment of a misconfiguration based on real-world conditions, such as the current availability of exploits or the importance of affected assets.

Still, CCSS also presents some limitations, as described in [18]. One of the main challenges is its relatively limited adoption compared to more established frameworks such as CVSS which can hinder integration with widely used tools and platforms. The scoring process may require detailed contextual knowledge and manual input, reducing automation potential and introducing subjectivity in certain cases. Another weakness lies in the complexity of properly assessing environmental factors, which can vary significantly between organizations and may be difficult to accurately quantify.

Despite these drawbacks, CCSS remains a promising framework for enhancing the visibility and prioritization of configuration-related risks.

3.3.3 Common Misuse Scoring System (CMSS)

The Common Misuse Scoring System (CMSS) is a structured framework developed by NIST to quantify the severity of vulnerabilities arising not from software flaws or misconfigurations but from misuse of legitimate software features, such as phishing via email attachments or abused administrative functions. CMSS adapts the familiar three-tier approach of CVSS to capture how misuse vulnerabilities differ from traditional flaws. Base metrics assess intrinsic properties, such as whether misuse can be leveraged actively or only passively, the complexity of the action, the required privilege levels, and the direct impact on confidentiality, integrity, or availability. Temporal metrics then introduce situational dynamics, including the availability of attack mechanisms, the readiness of mitigations, and the confidence in reports. Environmental metrics customize the score to reflect organizational priorities and the broader deployment context. As described in [19], CMSS provides standardized, quantitative severity scores (0–10) that allow consistent prioritization and informed risk management.

Strengths and weaknesses

One of its main advantages is its ability to address a category of vulnerabilities that are often overlooked by traditional frameworks like CVSS: misuses of legitimate software features rather than outright technical flaws. This makes this framework particularly useful in scenarios involving insider threats, social engineering, or insecure default behaviors, where attackers exploit functionalities in unintended ways. Furthermore, CMSS retains a familiar structure which promotes usability for professionals already accustomed to CVSS like systems.

However, CMSS also presents notable limitations. It is less mature and less widely adopted than CVSS, resulting in fewer community contributions, tools, and datasets that support it. [20] explains that its scoring criteria can be more subjective, especially when evaluating misuse complexity or potential impact, which may lead to inconsistent scores across analysts or organizations. The lack of automation support and integration in popular vulnerability management systems further hinders its practical applicability.

Still, CMSS fills an important gap by enabling structured assessment of non-technical vulnerabilities, and with further refinement and broader adoption, it has the potential to significantly enhance misuse-focused risk evaluation.

3.3.4 Exploit Prediction Scoring System (EPSS)

The Exploit Prediction Scoring System (EPSS) is a data-driven framework designed to estimate the likelihood of exploitation of publicly known software vulnerabilities. Developed by the Forum of Incident Response and Security Teams (FIRST), EPSS

complements existing vulnerability scoring systems by focusing specifically on the probability that a vulnerability will be exploited in the wild, rather than solely on its technical severity.

The system operates by analyzing a variety of features associated with each vulnerability, which are collected from multiple open-source and commercial datasets. These features include static attributes such as CVSS base metrics, as well as dynamic signals like the time since disclosure, the number of references in public databases, and the availability of proof-of-concept exploits or weaponized code. EPSS also integrates external signals, such as mentions in exploit databases.

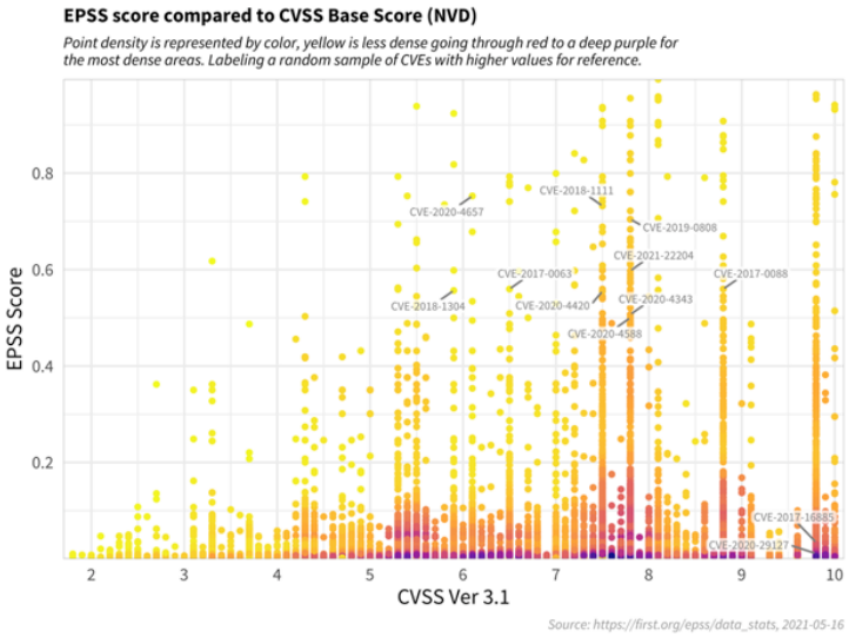

Figure 3.3: EPSS and CVSS comparison

As detailed by [21], EPSS uses machine learning model which is trained on historical data mapping CVEs to observed exploit activity gathered from real-world telemetry, such as honeypots or intrusion detection systems. The model produces a score between 0 and 1 for each vulnerability, which can be interpreted as the probability that the vulnerability will be exploited in the wild in the next 30 days. For example, a score of 0.85 would indicate an 85% likelihood of exploitation in that time frame.

Importantly, EPSS scores are updated daily, allowing them to reflect the dynamic nature of the threat landscape. This temporal granularity enables security teams to adapt quickly to emerging threats. Moreover, since EPSS is continuously retrained and improved based on new data, its predictive performance tends to increase over

time, making it a valuable complement to more static assessment tools.

In practice, organizations can use EPSS to supports risk based asset management by helping to align vulnerability mitigation efforts with business priorities, minimizing operational disruptions. In large-scale infrastructure, such as cloud environments or enterprise networks, EPSS can significantly improve incident response readiness by identifying likely exploit targets in advance.

Strengths and weaknesses

As said before and highlighted in [22] one of its primary advantages consists of its data-driven and probabilistic nature, this enables organizations to prioritize vulnerabilities more effectively, focusing resources on those with the highest exploitation risk, even if their CVSS severity scores are not particularly high. It is also freely available, making it accessible and timely for integration into security operations.

EPSS also presents some limitations. First, it is entirely dependent on historical and publicly available data, which may not capture emerging or targeted threats that haven't been widely observed yet. This can lead to underestimating the risk of newly disclosed or sophisticated zero-day vulnerabilities. Moreover, EPSS provides a single numerical probability without contextual information about the asset's criticality or environment, meaning it lacks the depth of analysis offered by environmental or temporal factors in other scoring systems like CVSS.

As a result, while EPSS is highly effective as a supplementary tool, it is best used in conjunction with other frameworks to provide a more comprehensive assessment of cyber risk.

3.3.5 Security Content Automation Protocol (SCAP)

The Security Content Automation Protocol (SCAP) is a suite of specifications developed by the National Institute of Standards and Technology (NIST) to standardize the communication and evaluation of security vulnerabilities and configuration issues in IT systems. SCAP is designed to support automated vulnerability management, measurement, and policy compliance evaluation across systems and organizations. As detailed in [23] and [24] SCAP combines multiple standards, such as the Common Vulnerabilities and Exposures (CVE), Common Configuration Enumeration (CCE), Common Platform Enumeration (CPE), Common Vulnerability Scoring System (CVSS), and many others into a unified framework.

When an SCAP-compatible tool is used to assess a system, it first uses CPE to identify the platform, then references CVE and CCE entries relevant to that platform. It applies XCCDF-defined benchmarks to test configurations using OVAL definitions. Results are scored using CVSS and reported back in a standardized format. This process supports regular, repeatable, and auditable assessments,

significantly reducing manual effort and inconsistencies in security evaluations.

SCAP is widely used in enterprise and government environments to automate the detection of misconfigurations, benchmark compliance against security baselines, and report on system security posture. It plays a key role in regulatory compliance, including the U.S. Federal Information Security Management Act.

Strengths and weaknesses

SCAP presents several strengths that make it a powerful tool for automated security assessment and compliance management. One of its main advantages lies in its standardization and interoperability: by combining well-defined components it ensures that different security tools can communicate using a shared language. This significantly enhances the consistency, repeatability, and scalability of vulnerability assessments across diverse systems. It also supports automation, reducing the manual effort required, which decreases the risk of human error.

Even so, SCAP has limitations: its reliance on up-to-date vulnerability and configuration databases means that outdated feeds can reduce accuracy and relevance. SCAP, also, tends to focus on known vulnerabilities and predefined checks, making it less effective against zero-day threats or complex attacks that do not conform to simple signature-based detection. while SCAP excels in assessing compliance, it may not provide deep contextual insights or dynamic risk prioritization unless integrated with complementary frameworks or real-time analytics tools.

3.3.6 Practices for Risk Indicator Scoring Maturity Assessment (PRISMA)

PRISMA is a qualitative framework developed by NIST for assessing the maturity of an organization's risk management practices, particularly in cybersecurity contexts. Originally designed in [25] it evaluates the extent to which risk management practices are institutionalized and consistently applied across an organization by examining various key areas such as risk framing, risk assessment, risk response, and risk monitoring. Rather than focusing solely on technical vulnerabilities or threat likelihoods, PRISMA emphasizes the process maturity of cybersecurity governance, making it a valuable complement to quantitative scoring models like CVSS or EPSS.

It uses a qualitative assessment model based on five maturity levels: *Ad Hoc, Defined, Consistently Implemented, Measured and Managed, and Optimized*. Each level represents a stage in the institutionalization and effectiveness of risk management processes. PRISMA works by first identifying specific risk management functions, for each of which a set of indicators or best practices is defined. Evaluators then assess how well the organization has adopted each practice, scoring it

according to its maturity level. This scoring is done using predefined criteria that describe behaviors, documentation, and repeatability of actions within each level.

The assessment is typically carried out through structured interviews, document reviews, and policy analysis, often involving stakeholders from multiple departments. The results are aggregated into a maturity profile that helps pinpoint areas of strength and weakness, guiding strategic improvements.

Strengths and weaknesses

[26] explains that PRISMA structure and process-oriented approach provides organizations with a clear and repeatable methodology for assessing the maturity of their risk management practices. By focusing on the *how*how rather than the *what*, PRISMA enables organizations to identify systemic gaps and inefficiencies in their risk governance processes, offering a roadmap for continuous improvement. It also aligns well with broader risk management and compliance frameworks, such as the NIST Cybersecurity Framework, promoting integration across different risk domains.

Its qualitative nature means that it relies heavily on subjective assessments, which can vary depending on the experience and interpretation of the evaluators. While it is effective at measuring maturity and consistency, PRISMA does not directly assess the technical severity of vulnerabilities or the likelihood of specific cyber threats, like CVSS and EPSS.

3.3.7 Cybersecurity Framework (CSF)

The NIST Cybersecurity Framework (CSF), first released in 2014 by the National Institute of Standards and Technology, has become a widely adopted standard to improve cybersecurity risk management in both the public and private sectors. Originally developed to improve critical infrastructure security and resilience, CSF provides a flexible and cost-effective framework that organizations of any size and industry can adopt. As [27] describe its functionality is structured around five core functions:

- Identify
- Protect
- Detect
- Respond
- Recover

which represent the lifecycle of an organization's cybersecurity posture. These functions provide a strategic view of how organizations can understand and address cybersecurity risks.

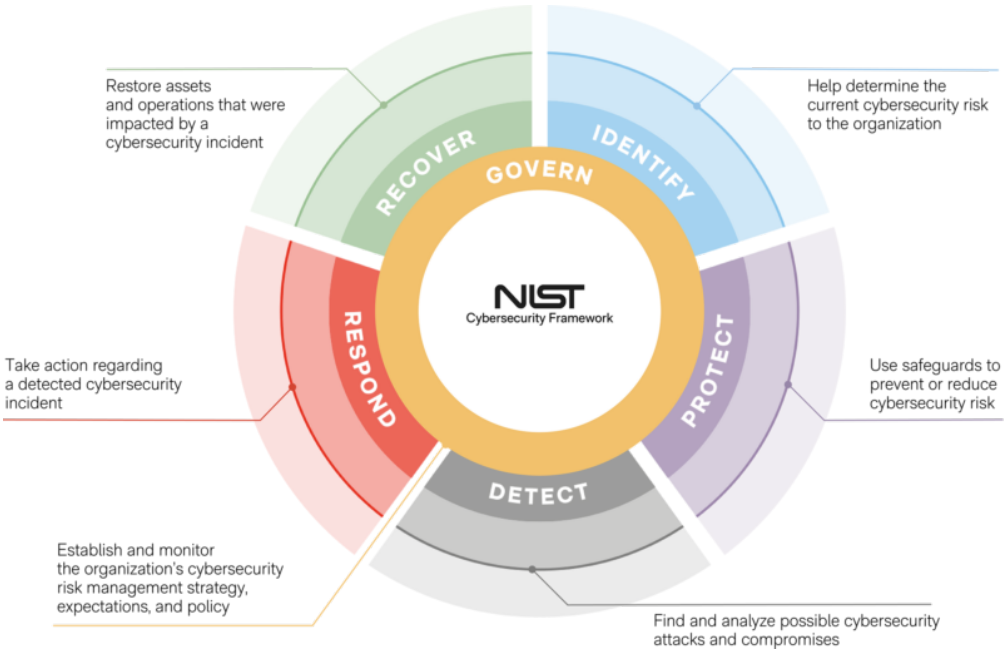

Figure 3.4: CSF main functions

Each function is divided into categories and subcategories, which detail specific objectives and desired outcomes. For example, under Identify, categories may include asset management, governance, and risk assessment. Under Protect, examples include access control, data security, and awareness training.

The CSF is implemented through three key components:

- **The Framework Core**: which includes the functions, categories, and subcategories.
- **The Implementation Tiers**: which describe the degree to which an organization's cybersecurity practices exhibit the characteristics defined in the Framework
- **The Framework Profile**: which represents the alignment of the Framework Core with an organization's specific business requirements, risk tolerance, and resources. Organizations can define a "Current Profile" and a "Target Profile" to identify gaps and prioritize improvements.

It is commonly used to assess current cybersecurity capabilities, set improvement goals, and communicate risks both internally and externally. By aligning

cybersecurity activities with business requirements and regulatory obligations, the CSF helps integrate cybersecurity risk management into an organization's overall risk management strategy.

Strengths and weaknesses

The CSF is known for its flexibility and scalability: its design is built to be adaptable to organizations of any size, sector, or cybersecurity maturity level, making it especially useful for entities that lack the resources of larger institutions. its risk-based and outcome-focused approach allows organizations to prioritize security investments based on their unique business needs and risk tolerance. The framework also promotes alignment with existing standards, providing a common language for security practices and facilitating regulatory compliance.

A key challenge lies in its lack of specificity, which, while offering flexibility, may result in vague guidance for organizations seeking concrete implementation steps, especially those with less security expertise, as written in [28]. The voluntary nature of the framework may lead to inconsistent adoption, particularly in industries not bound by cybersecurity regulations. Another limitation is that the CSF, being primarily focused on process and governance, does not provide detailed technical controls or address emerging threats with the same granularity as more specialized frameworks.

3.3.8 Stakeholder-Specific Vulnerability Categorization

The Stakeholder-Specific Vulnerability Categorization (SSVC) framework is a decision-making model introduced to help organizations prioritize vulnerability remediation based on their specific operational context and mission impact. As illustrated in [29], SSVC addresses the limitations of traditional scoring systems such as CVSS by incorporating factors that go beyond technical severity, such as the status of exploitation, public impact, and the criticality of the affected system within the organization. It operates through a *structured decision tree model* that, rather than relying on a numerical score, guides users through a set of branching decisions, each corresponding to a key factor relevant to operational risk and organizational mission.

The core decision points in the SSVC tree include:

- **Exploitation Status**: whether the vulnerability is being actively exploited in the wild.
- **Exposure**: whether the vulnerable system is exposed to potential attackers.
- **Technical Impact**: the potential severity of the vulnerability from a system integrity, availability, or confidentiality point of view.

- **Mission Prevalence**: how critical the affected asset is to the organization's specific operational or mission functions.
- **Public Well-being Impact**: whether exploitation could harm the public, especially in healthcare or public infrastructure.

Each of these factors is assessed using clearly defined criteria, leading to different leaf nodes in the decision tree, which translate into one of four actions:

- **Track**: monitor the vulnerability but do not take immediate action.
- **Assess**: perform further investigation and contextual analysis.
- **Attend**: delay action due to low urgency or operational limitations.
- **Act**: prioritize immediate remediation.

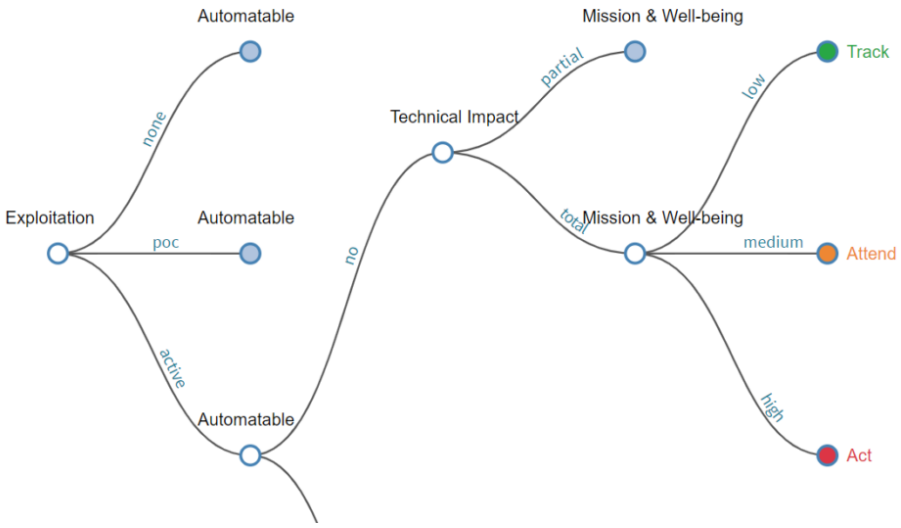

Figure 3.5: Portion of the SSVC structured decision tree

Strengths and weaknesses

As remarked in [30] SSVC considers organizational mission impact, safety implications, and exploit availability when prioritizing vulnerabilities, allowing for more accurate and relevant decision-making. This stakeholder-oriented approach enables different entities, such as asset owners, vendors and coordinators, to make specific

remediation decisions based on their unique perspectives and responsibilities. Additionally, the decision-tree structure offers a clear, repeatable, and auditable process that supports transparency and consistency across assessments.

Unlike scoring-based models like CVSS, SSVC does not generate a numerical risk score, which can be a drawback for organizations seeking integration with automated tools or quantitative risk dashboards. Its decision-making process also relies heavily on qualitative input, which can introduce subjectivity and requires a certain level of expertise and manual effort to apply effectively. Moreover, SSVC is relatively new and less widely adopted, which may hinder interoperability with existing systems and frameworks.

3.3.9 Synthesis of Findings

Framework	Strengths	Weaknesses
CVSS	Widely adopted and standardized Quantitative scoring system Supports risk prioritization	May oversimplify risk Lacks real-world context Static over time
CCSS	Tailored to configuration issues Adds contextual metrics Aligns with CVSS structure	Still experimental Limited adoption Few available tools
CMSS	Focuses on configuration vulnerabilities Useful for system-specific risk analysis Extends CVSS for configs	Complex implementation Lacks broad support Requires detailed input
EPSS	Data-driven exploit prediction Supports prioritization by likelihood Updated daily	Black-box model Lacks transparency Cannot be manually tuned
SCAP	Standardized automation Interoperable with multiple tools Reduces manual effort	Complex to implement Requires frequent updates Not intuitive for non-experts
PRISMA	Holistic and flexible assessment Useful for gap analysis Aligns with policy and strategy	Qualitative, not numerical Time-consuming Subjective scoring
CSF	Flexible and widely recognized Supports continuous improvement Aligns with business objectives	Lacks detailed technical guidance Requires adaptation effort Can be high-level for OT use
SSVC	Context-aware decision making Stakeholder-specific guidance Easy-to-use decision trees	No quantitative score Requires manual input Limited tool integration

Table 3.1: Strengths and weaknesses of the analysed frameworks

3.4 Relevant Research and papers

In addition to the analysis of existing frameworks, several research papers have been examined to provide theoretical grounding and practical context to the development of this work. These articles cover a wide range of topics relevant to cyber risk assessment, including threat modeling, security automation, and decision support systems.

Each of them offers valuable perspectives either by proposing novel methodologies or by critically evaluating existing approaches.

Their content has been fundamental for the design of the proposed model.

3.4.1 Quantifying the Impact Propagation of Cyber Attacks using Business Logic Modeling

The article [31] offers an innovative perspective on analyzing the spread and effects of cyber attacks within the complex infrastructure of an organization.

The authors propose a framework that integrates business logic with technical dependencies, allowing for a detailed assessment of how disruptions in one component can ripple through interconnected systems and processes. This approach goes beyond traditional vulnerability analysis by considering not only individual asset risks but also the broader systemic impact based on the organization's operational workflows.

A key element of the study is the examination of network connections and the exchange of data packets as a means to understand the underlying communication patterns between components. By analyzing these data flows, the model calculates network centrality metrics, which identify the most influential nodes within the infrastructure: those whose compromise or failure would have the greatest cascading impact.

	A1	**A2**	**A3**	**A4**
A1	0.0	0.56	0.72	0.64
A2	0.52	0.0	0.28	0.36
A3	0.26	0.10	0.0	0.0
A4	0.22	0.34	0.0	0.0

Table 3.2: Matrix showing the exchange of data packets

Unlike conventional approaches that rely primarily on static network maps or asset inventories, this dynamic evaluation based on real-time communication offers a more accurate representation of critical points within the network. Such insight is crucial for prioritizing defense mechanisms and efficiently allocating cybersecurity

resources.

This methodology supports a deeper understanding of how cyber threats can propagate in operational environments, enabling organizations to better predict and contain potential attack vectors by focusing on the components that serve as central hubs in their network ecosystem.

3.4.2 Cyber Risk Quantification: Investigating the Role of Cyber Value at Risk

The article [32] proposes a financial approach to the evaluation of cyber risks. Drawing inspiration from the well-established concept of Value at Risk (VaR) in finance, the authors introduce the notion of Cyber Value at Risk (CyVaR) as a method to estimate the potential monetary loss resulting from cyber incidents. This approach shifts the focus from technical metrics to economic consequences, providing decision makers with a clearer picture of the financial exposure of an organization to threats. A major challenge in applying this methodology lies in the scarcity and inaccessibility of reliable data needed for accurate modeling, such as historical incident costs, internal revenue structures, dependency graphs among digital assets, and the specific impact on business processes. These data points are often proprietary, fragmented, or simply unavailable due to confidentiality, making the practical implementation of CyVaR models complex and organization-specific.

3.4.3 A system to calculate Cyber Value-at-Risk

The article [33] presents a structured methodology for assessing the financial exposure of organizations to cyber threats. The proposed system models cyber risk through a combination of threat intelligence, asset valuation, vulnerability data, and loss distribution estimation, enabling decision makers to simulate various attack scenarios and estimate their economic impact. This allows for the production of risk metrics that are not only tailored to the organization's specific context but also useful for communication with nontechnical stakeholders such as executives and board members.

Chapter 4

Risk Evaluation Model Design

This chapter presents the main methodology steps that have been taken in this work to design and build the first prototype of the reporting engine, deducing possible risk scores depending on the architectures, vulnerabilities and misconfiguations discovered during the reconnaissance phase (which is not included in the scope of the work as mentioned in the previous chapters).

This chapter will begin by providing a high-level overview of the engine's design, offering the reader a clear understanding of the overall architecture and the interactions between the various modules.

It will then look into a more detailed explanation of each module's specific role, emphasizing the technical decisions made during the implementation process.

4.1 Design of the engine

The goal of the engine is to receive as input the full description of the network that has been selected as target of evaluation, extract all the information needed regarding the topology, the vulnerabilities, the possible attack paths, and computing a score for each host representing its threat level and the possible damage a malfunction could cause.

As discussed in the chapter 3, the method that has been chosen for generating the scores is a combination of existing frameworks. Through these, it becomes possible to assess the various factors that collectively contribute to increased exposure to cyber attacks, factors that, if considered in isolation, would fail to provide a complete understanding of the overall security situation.

With regard to the input, a dedicated structure has been designed to consolidate all the information collected during the reconnaissance phase into a single and

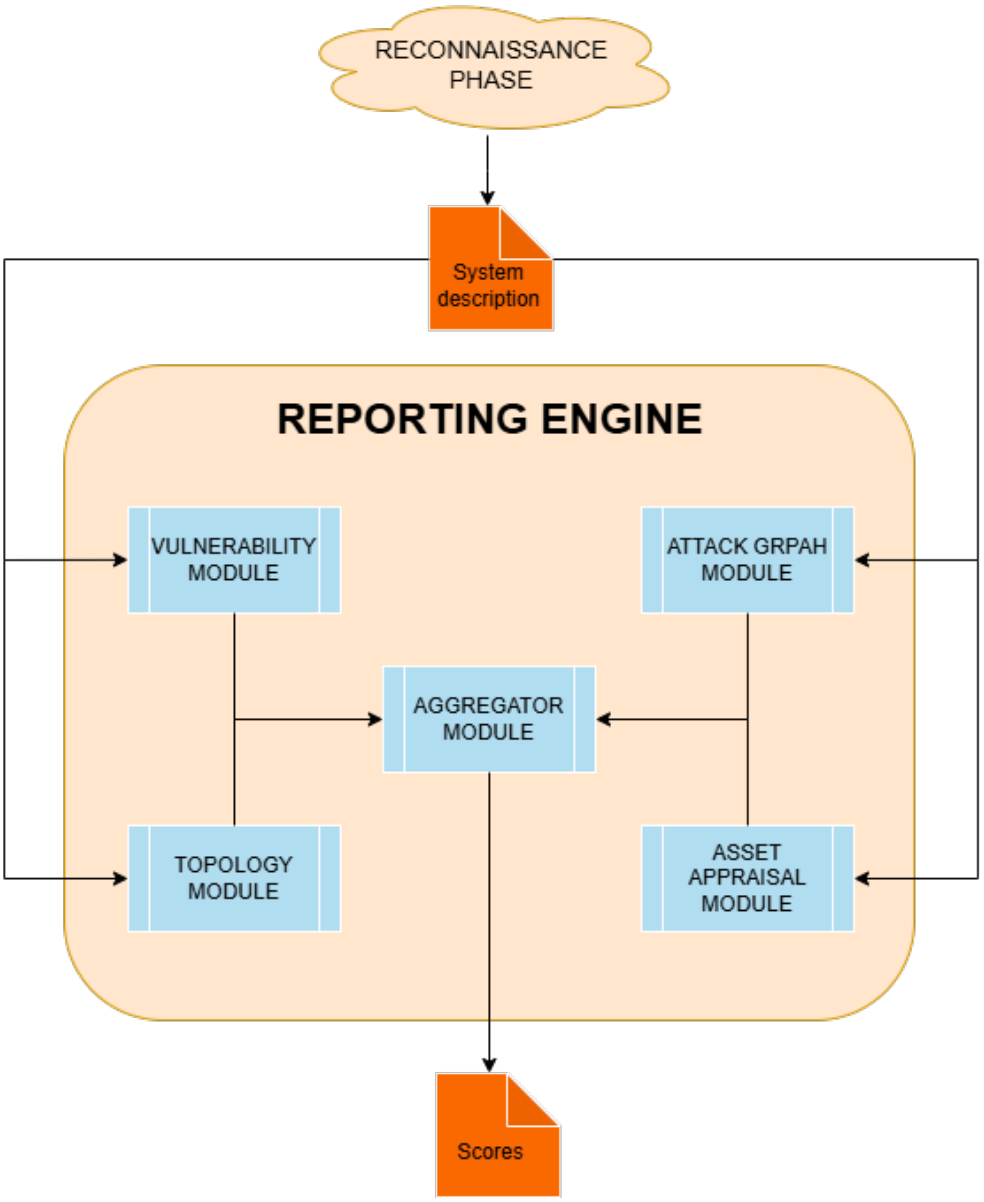

Figure 4.1: General architecture of the proposed engine

standardized format. This approach facilitates future automation of data generation and ensures that the information can be easily accessed by any component or resource that requires it.

From a design point of view, the engine has been split into four distinct modules:

- **Vulnerability Module**: it is responsible for receiving and processing detailed information about known vulnerabilities affecting system components. Collect

data such as CVE identifiers, severity metrics, affected assets, and exploitability indicators. With this information, the module calculates a risk score for each host, reflecting its potential impact on the overall infrastructure.

- **Attack Graph Module**: it is designed to analyze and evaluate attack graphs, which model the potential paths an attacker could take within a network by chaining together vulnerabilities and system misconfigurations. It processes the input graph to identify critical nodes and possible lateral movement strategies. Determining the complexity and likelihood of each attack path, it generates a score that reflects the overall risk exposure of the hosts based on their structural weaknesses. This evaluation assist in understanding how threats propagate and in identifying strategic points for defense reinforcement.

- **Topology Module**: it focuses on analysing the network topology to assess how the structure and interconnections between hosts influence the overall security posture. Examines how devices are linked, the flow of communication, and the roles of individual nodes within the infrastructure. By identifying central or highly connected hosts, potential bottlenecks, and isolated segments, it uncovers critical points that may serve as key targets or weak spots in the event of an attack. Based on this analysis, it produces a score that reflects the topological exposure of the network, providing insights into how architectural decisions impact risk.

- **Asset Appraisal Module**: it is responsible for evaluating the criticality of each asset within the network based on its operational and economic value. It takes into account factors such as the asset's role in business processes, its sensitivity, and the potential impact its compromise could have on the organization's functionality and finances. The module helps prioritize protection efforts and risk mitigation strategies classifying assets according to their importance. At the end of the evaluation, it assigns a score to each asset, representing its level of criticality within the overall infrastructure.

As the final stage of the evaluation process the different outputs are integrated and synthesized into a single, coherent risk score by the **aggregator module**. The module ensures that no single factor is considered in isolation, allowing for a more balanced and accurate representation of risk with the aggregation of these diverse perspectives. The resulting global score reflects not only the presence of threats but also their potential impact within the context of the specific infrastructure.

Figure 4.1 visually represents the architecture described above. It is easy to see how the previously stated modules, which are coloured in light blue, link together, and also what are the required inputs and produced outputs, which correspond to the orange files in the image.

4.2 Description of the vulnerability module

The vulnerability module plays a central role in the overall risk assessment model, as it quantifies the degree of exposure introduced by known security weaknesses on each individual host.

Evaluating vulnerabilities is a critical component in the broader context of cybersecurity, they represent weaknesses in a system that, if exploited, can lead to unauthorized access, data breaches, or disruption of essential services. Organizations that analyse vulnerabilities inside their network gain insight into which components of their infrastructure are most at risk and require immediate attention. Their evaluation helps anticipate potential attack vectors and supports the implementation of proactive defense mechanisms. Without this step, security strategies would lack focus and systems would remain exposed to known threats, increasing the likelihood of successful cyberattacks and their associated operational and financial impacts.

4.2.1 Formula Definition

The scoring formula adopted in this module is designed to provide a more nuanced and context-aware assessment of the risk posed by known vulnerabilities. It is calculated taking in consideration all vulnerabilities for each host with the expression:

$$\frac{20}{\pi} \cdot \arctan\left(\sum_{i=1}^{n}\left(\mathrm{CVSS}_i^{1.5} \cdot \mathrm{EPSS}_{\mathrm{percentile}_i} \cdot \mathrm{Attack\_Rate}_i\right)\right)$$

Using the arctangent function in the vulnerability scoring formula helps normalize and compress large input values into a manageable and interpretable range. The raw summation of products may produce very large numbers. Without normalization, these values would be difficult to compare across different systems or environments. This curve asymptotically approaches 10, which allows us to limit the output, effectively scaling it to a range between 0 and 10.

This ensures that no matter how high the input grows, the final score remains within a defined and comparable scale. Furthermore this non-linear behavior allows the model to be more sensitive to changes when the input values are low—where small differences matter most—while becoming more stable as values grow larger helping in balancing detail in the scoring process.

The CVSS (Common Vulnerability Scoring System) score represents the intrinsic severity of the vulnerability, taking into account factors such as exploitability and potential impact. In this formula, the CVSS score is raised to the power of 1.5 to amplify the effect of more severe vulnerabilities and give them more weight in the final risk calculation. The non-linear weighting ensures that high-severity issues

disproportionately influence the overall score, aligning with the principle that more critical vulnerabilities pose a significantly greater threat.

The EPSS (Exploit Prediction Scoring System) introduces a dynamic element that estimates the probability that a given vulnerability will be exploited in the wild. The choice of using the percentile has been made because the EPSS raw score ranges between 0 and 1, but it is not uniformly distributed.

The percentile, on the other hand, represents a relative ranking among all known vulnerabilities. For example, a vulnerability in the 90th percentile is more likely to be exploited than 90% of others, this gives a clearer and more interpretable context. Many vulnerabilities may have similar low scores, which can make comparisons difficult. In fact, with incorporation of it, the formula shifts from a purely theoretical risk assessment to one that reflects real-world threat activity. This helps prioritize vulnerabilities not only based on severity but also on how likely they are to be targeted by attackers.

Finally, Attack rate is the percentage of attacks derived from the vulnerability and represents empirical data that indicate how often a specific vulnerability has the potential of being exploited in the real world scenario of the network. This metric adds a valuable layer of realism to the evaluation by grounding the score in observed threat behavior, rather than relying solely on theoretical exploitability or predictive models. In practice, not all vulnerabilities, even those considered highly exploitable, are equally targeted by attackers. Some may be technically severe, but rarely used in the wild, while others with moderate scores might be exploited frequently due to widespread deployment or ease of automation.

This factor was intentionally integrated into the formula to improve the comparability and relevance of the risk score in different operational contexts. With the inclusion of an empirical exploitation rate, the model becomes more adaptable and realistic, aligning better with scenarios where decision-makers must prioritize based not only on potential severity, but also on actual threat trends.

In the context of this work the parser is implemented as a Python script that receives as input the list of the existing CVE with their respective information.

4.2.2 Definition of input data

The model must provide a way for representing the potential vulnerabilities discovered during the reconnaissance phase. For the purpose of this work, vulnerabilities are linked to a host. The input data is structured as a JSON list, where each entry represents a distinct vulnerability. Each item in the list includes several key attributes: a unique identifier (id) for internal reference, the corresponding Common Vulnerabilities and Exposures (cve) code, the CVSS score indicating the severity of the vulnerability, the EPSS percentile reflecting the likelihood of exploitation in the wild, and finally, the number of known attacks that have originated from that

specific vulnerability. This structured format allows both human readability and efficient machine processing, facilitating automation and integration with other modules in the risk assessment pipeline. Outside the list another field contains the total amount of attacks that use vulnerabilities as a starting point or as a way to advance in their accomplishment.

Here is an example of input data:


```
"total_attacks": 6,
"vulnerabilities": [
    {
      "ID": 1,
      "CVE": "CVE-2022-25888",
      "host": ["PLC1"],
      "score": 7.5,
      "EPSS": 0.51,
      "attacks": 3
    },
    {
      "ID": 2,
      "CVE": "CVE-2022-25888",
      "host": ["PLC2"],
      "score": 7.5,
      "EPSS": 0.51,
      "attacks": 2
    },
    {
      "ID": 3,
      "CVE": "CVE-2025-49710",
      "host": ["HMI1", "HMI2", "HMI3"],
      "score": 9.8,
      "EPSS": 0.8,
      "attacks": 3
    },
]
```


4.2.3 Definition of output data

The output of the formula is a numerical value that is normalized and bounded within the range [0, 10]. The result is a score that behaves similarly to traditional vulnerability metrics like CVSS but benefits from a more refined and realistic scaling. Such a format makes the final value easily interpretable and compatible with the other scoring modules of the model.

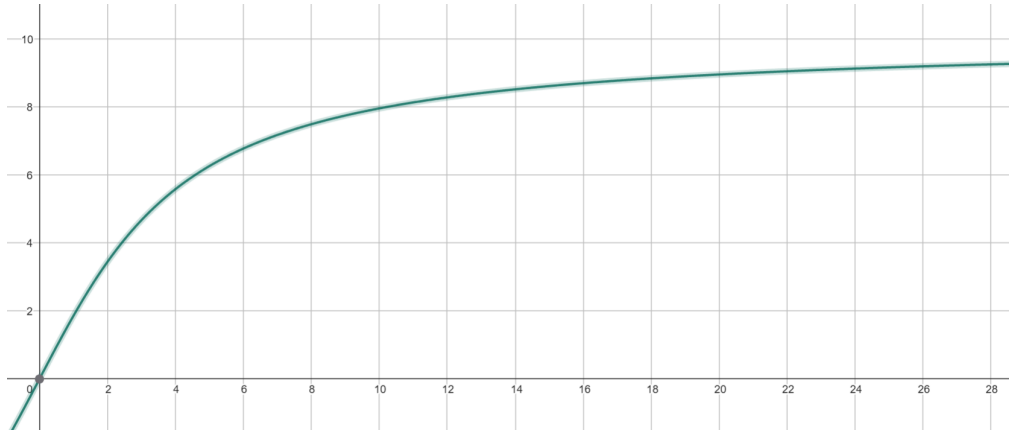

Figure 4.2: Graphical representation of the vulnerability scoring formula

Figure 4.2 illustrates how the vulnerability score increases as the combined input values grow.

4.3 Description of the attack graph module

The Attack Graph Module is fundamental in assessing the potential pathways an attacker could exploit within a networked infrastructure.

Evaluating attack paths provides critical insights into how threats can propagate through a system. Rather than focusing solely on isolated vulnerabilities, it allows defenders to understand the sequence of steps an attacker could take to move from an initial point of compromise to more sensitive or critical components, revealing the dependencies between systems and highlighting how even low-severity vulnerabilities, when chained together, can lead to significant damage. Organizations can prioritize defenses more effectively, strengthen weak links in their infrastructure, and deploy targeted mitigation strategies that disrupt potential attack chains before they can be exploited.

4.3.1 Formula Definition

The scoring formula implemented in this module aims to deliver a more detailed and context-sensitive evaluation of the risk associated with known attack paths. It is computed for each host through the following expression:

$$\text{Score} = \frac{20}{\pi} \cdot \arctan\left(\frac{x}{\frac{N_{\text{AG}}}{N_{\text{total}}}}\right) \quad \text{where} \quad x = \sum(\#\text{children} + \#\text{parents}) \cdot f$$

$$\text{with } f = \begin{cases} 1.2, & \text{if the node is an entry point or a leaf node} \\ 1, & \text{otherwise} \end{cases}$$

As in the previous module 4.3, the use of the arctangent function in this formula serves the purpose of normalizing the output and introducing a nonlinear scaling effect ensuring that the result remains within a bounded range, facilitating comparison with other modules and maintaining interpretability.

The value x in the formula represents the cumulative connectivity of a given host within the attack graph, reflecting both the number of attacks that target the host and those that originate from it. Specifically, it is calculated as the sum of all incoming and outgoing edges and capturing how central the host is within the overall flow of potential attack paths. This dual perspective allows the model to quantify not only how exposed the host is to external threats, but also how likely it is to serve as a stepping stone in lateral movement through the network.

The parameters N_{AG} and N_{total} are fundamental in the scoring formula because they introduce a mechanism for contextual normalization. Specifically, N_{AG} denotes the number of the attack graph nodes that involve the specific host we are analysing, while N_{total} refers to the total number of nodes present in the graph.

Their ratio quantifies the extent to which each host of the network is exposed or susceptible to attacks, offering a proportional metric.

This normalization factor is especially important when applying the model across varied network environments with differing sizes and complexities Without it, the resulting score could be disproportionately influenced by the absolute number of connections or nodes, leading to inconsistencies when comparing networks of different scales.

Thanks to this, the module, ensures that the score reflects not just the internal structure of the attack graph, but also how significant it is in the context of the entire system. It also enhances the model's versatility and reliability, enabling its application to both small and large-scale environments while preserving the comparability and interpretability of the results.

Finally applying a modifier to the entry and leaf nodes of an attack graph is essential, as these nodes represent the most strategically significant positions in potential attack paths.

Entry nodes are the first points an attacker might exploit to access the system, often reflecting vulnerabilities at the network perimeter.

Leaf nodes, in contrast, usually represent critical assets or objectives, such as sensitive data or key services, that attackers aim to reach.

By assigning a higher weight to these nodes, the model highlights their greater importance within the overall structure. This approach allows the scoring formula to better reflect real-world risk by giving more influence to nodes that either initiate or complete an attack chain. As a result, the module helps prioritize defensive

measures where they can be most effective, enhancing both the accuracy and practicality of the risk assessment.

4.3.2 Definition of input data

The input data for the attack graph module consists of a structured representation of the network's potential attack paths, commonly referred to as an attack graph. This graph is composed of nodes representing individual hosts or network components, and edges that denote possible exploit-based transitions between them. The graph is generated by the formal verification engine developped in [34] and illustrated in Figure 3.1 using information gathered during the reconnaissance phase, such as network topology, identified vulnerabilities, and known exploit chains. It provides a comprehensive view of the system's security posture by capturing how an attacker could move laterally through the infrastructure.

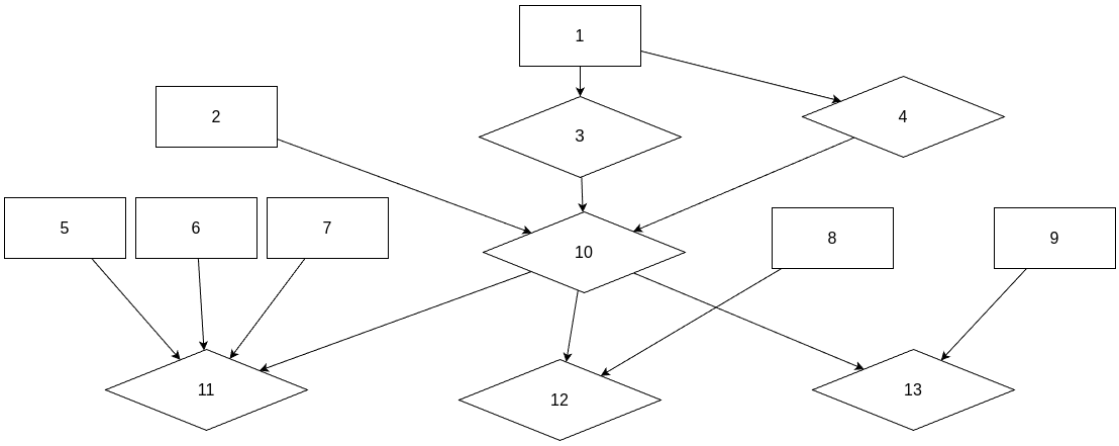

Figure 4.3: Example of an attack graph

To simplify the design and the implementation with the python script the graph has been transformed into a JSON structure with this shape:

```
{
    "nodes": [
      {"ID": 1, "host": ["PLC1"], "fathers": [], "sons": [3, 4]},
      {"ID": 2, "host": ["ScadaPC"], "fathers": [], "sons": [13]},
      {"ID": 3, "host": ["ScadaPC"], "fathers": [1], "sons": [13]},
      {"ID": 4, "host": ["ScadaPC"], "fathers": [1], "sons": [13]},
      {"ID": 5, "host": ["PLC1"], "fathers": [], "sons": [15]}
      ]
}
```

Node ID	Node label
1	vulExists(fortim,13,customPacketFilter,remoteExploit,accessControlBypass)
2	vulExists(scadaPC,9,ignitionPortal,remoteExploit,privEscalation)
3	accessControlBypass(companyPC,scadaPC,https,443)
4	accessControlBypass(scadaPC,scadaPC,https,443)
5	vulData(plc1Data,10,unencrypted,sniffing)
6	vulLinkProtocol(plcLan,5,ethernet,adjacent,eavesdropping)
7	vulE2EProtocol(plc1,historian,6,mqtt,1883,adjacent,eavesdropping)
8	vulExists(plc1,1,opcuaServer,remoteExploit,dos)
9	vulLinkProtocol(plcLan,4,arp,adjacent,impersonateDst)
10	execCode(scadaPC,admin)
11	accessDataFlow(attacker,plc1Data,view)
12	dos(attacker,plc1)
13	mitmLink(attacker,hmi1,plc1,scadaPC)

Table 4.1: Node labels corresponding to Figure 4.3

4.3.3 Definition of output data

The output of the formula is a numerical value that is normalized and bounded within the range [0, 10]. This module provide a view of node exposure enabling security analysts to pinpoint which hosts present the greatest systemic risk and to align mitigation efforts strategically.

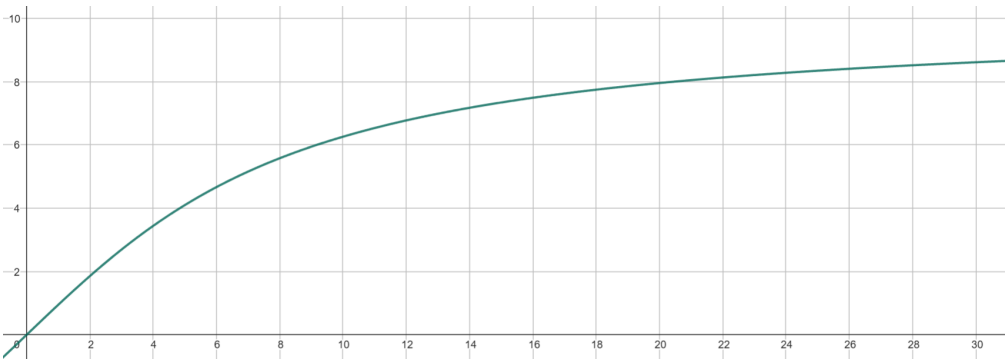

Figure 4.4: Graphical representation of the attack graph scoring formula

4.4 Description of the topology module

The Topology Module is a crucial in assessing the structural characteristics of the network under analysis. Its primary function is to evaluate how the arrangement and interconnectivity of hosts influence the exposure to cyber threats.

Unlike vulnerability or attack-path assessments, which focus on specific weaknesses or sequences of exploits, this module captures the broader layout of the network identifying central nodes, potential bottlenecks, and points of convergence that may represent critical risks. By analyzing the network's topology, the model gains awareness of which hosts serve as communication hubs or vital transit points, making them attractive targets for attackers. This structural perspective enables a more comprehensive risk evaluation and helps guide decisions about segmentation, redundancy, and protection of key assets.

4.4.1 Formula Definition

The formula implemented within the Topology Module is designed to quantify the strategic importance of each host based on its position and role within the network structure. It captures how central a node is in terms of communication flow, by considering the number of connections it maintains with other hosts. The goal is to translate abstract topological characteristics into a standardized, interpretable score that can be seamlessly integrated with the outputs of other modules in the risk evaluation framework. It is calculated for each host with this expression:

$$\text{Score} = \frac{20}{\pi} \cdot \arctan\left(\frac{n_c \cdot p_h}{n_h}\right)$$

where:

- n_c = number of hosts the node communicates with,
- p_h = number of packets sent and received by the host,
- n_h = total number of hosts in the network.

The parameter n_c the number of distinct hosts within the network that maintain a direct communication link with the evaluated host. It is a fundamental indicator of the host's topological centrality, providing insights into how structurally embedded the host is in the network's communication graph. A higher number of connections typically implies greater exposure to external inputs, increasing the likelihood that the host could be used as an entry point or pivot for lateral movement in a cyberattack. Additionally, highly connected hosts often perform coordination or gateway roles, making them critical targets from both an operational and security perspective. This metric therefore contributes to identifying strategic nodes whose compromise could have cascading effects across the system.

The number of packets p_h exchanged by a host (both inbound and outbound) is a proxy for its activity level and functional weight in the network. Hosts with high data throughput usually perform vital functions such as serving web content,

managing databases, or orchestrating internal services. As such, they are more attractive targets for attackers, both for initial compromise and for persistence strategies. Furthermore, a high traffic volume can also suggest a wider attack surface due to increased interaction with other systems, services, or external users. Incorporating this parameter allows the model to prioritize hosts not just by their position in the network but also by how actively they are being utilized, providing a richer and more accurate risk assessment.

The term n_h provides normalization relative to the overall size of the network, ensuring that the computed score remains comparable across different environments. or instance, a host connected to 10 other devices in a small network of 12 hosts is far more central than one with the same number of connections in a network of 1,000 hosts. By accounting for the total number of nodes, the formula avoids inflating the significance of metrics in large-scale infrastructures or underestimating them in small-scale ones. This normalization not only makes the metric scalable but also ensures that results remain consistent and interpretable across diverse organizational contexts and network architectures.

4.4.2 Definition of input data

The input to the topology module consists of a comprehensive set of data describing the communication flows between the hosts in the network. Specifically, it includes a list of all known connections established among the hosts, along with quantitative information on the number of packets exchanged in both directions for each connection. This data is typically collected through network monitoring tools or traffic capture systems during the reconnaissance phase and represented as a graph structure that models the communication relationships between the various hosts in the network. Each node in the graph corresponds to a specific host, while each edge represents an established communication link between two hosts. In addition to indicating the presence of a connection, each edge carries associated metadata, such as the number of packets exchanged between the nodes it connects.

This graph based representation allows the module to analyze both the structural position of each host—such as how many connections it has—and the intensity of those connections, deduced from the traffic volume.

By examining this graph, the module can identify highly connected nodes, detect patterns of dense communication, and highlight hosts that may play a critical role in the spread or containment of a cyber attack.

To facilitate seamless integration with the Python script responsible for computing the scores, the data representing the communication graph has been transformed into a more accessible and script-friendly structure. This transformation includes the option to select between two formats. Leaving this choice ensures compatibility with different processing needs and enhances the adaptability of the module to

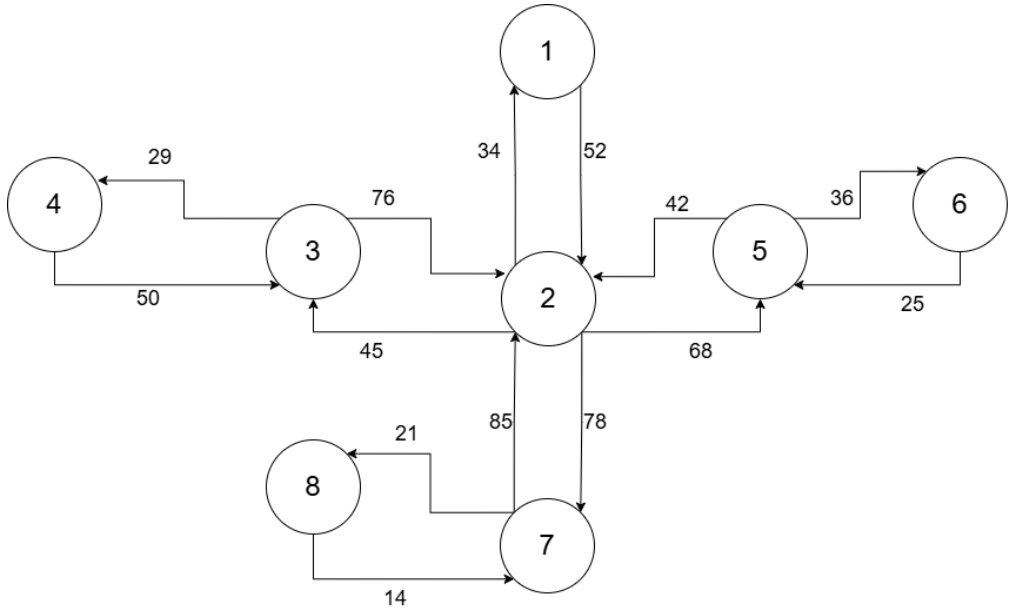

Figure 4.5: Example of the graph structure

various network topologies and computational requirements.

The first format used to represent the communication data is an adjacency matrix, where both the rows and columns correspond to the hosts in the network. Each cell at the intersection of a row and a column indicates the number of packets exchanged between the corresponding pair of hosts. This structure provides a clear and immediate overview of the intensity of communication across the entire network.

The matrix format is particularly effective in scenarios where a high volume of interactions is present, as it allows for efficient matrix based operations and simplifies the calculation of centrality or influence metrics. Its regular structure facilitates integration with various numerical libraries and tools commonly used in data analysis and machine learning workflows. In this case, the network scan used to collect communication data can also be performed at earlier stages, as the matrix format does not require real-time capture but simply a snapshot of the communication patterns between hosts.

	H1	H2	H3	H4	H5	H6	H7	H8
H1	0/0	1/5	1/5	1/5	2/5	3/4	2/2	2/4
H2	5/1	0/0	0/0	0/0	3/4	0/0	0/0	5/1
H3	5/1	0/0	0/0	0/0	3/4	0/0	0/0	4/1
H4	5/1	0/0	0/0	0/0	0/0	0/0	3/4	6/1
H5	5/2	4/3	0/0	0/0	0/0	0/0	0/0	0/0
H6	4/3	0/0	4/3	0/0	0/0	0/0	0/0	0/0
H7	2/2	0/0	0/0	4/3	0/0	0/0	0/0	0/0
H7	4/2	1/5	1/4	1/6	0/0	0/0	0/0	0/0

Table 4.2: Matrix of inter-host communication with inbound/outbound packet counts

The second input format adopted for the topology module relies on a real-time scanning approach using PyShark, a Python wrapper for TShark (the command-line version of Wireshark). In this configuration, network packets are captured live as they traverse the network, allowing the system to dynamically build and update a representation of host communications. Each captured packet is analyzed to extract essential information, such as source and destination addresses. This data is continuously aggregated into a graph structure that reflects the actual communication patterns observed during the monitoring period. This method is especially useful in environments where the network is highly dynamic or where historical data is unavailable, providing a contextual and up-to-date view of the topology and no prior scan is required, as all the necessary information is collected and processed in real time.

```
import pyshark

capture = pyshark.FileCapture()
for packet in capture:
    print(f"Source: {packet.src} -> Destination: {packet.dst}")
```

A big problem associated with network scanning is the risk of capturing a distorted or incomplete view of the infrastructure if the scan is conducted during a period of low or atypical activity. For instance, running a scan during off-peak hours, maintenance windows, off-peak hours, or moments of transient congestion might not reflect the true nature of communication flows, potentially leading to an inaccurate assessment of host centrality or connectivity.

To address this issue, two alternative approaches have been considered. The first involves conducting multiple scanning sessions at strategically chosen intervals throughout the day or over several days, thus capturing a broader range of typical network activity and reducing the likelihood of biased results.

The second strategy is to perform a single, prolonged scan that passively monitors network traffic over an extended period of time. This approach increases the chances of capturing representative communication behaviors and reduces sensitivity to temporary anomalies.

By implementing either, or both, of these solutions, the data feeding into the topology module becomes more robust, consistent, and reflective of the actual dynamics within the network.

4.4.3 Definition of output data

The output of the topology module is a normalized numerical score for each host, representing its level of centrality and communication intensity within the network. This score ranges from 0 to 10 and reflects how critical a node is in terms of network structure and data flow. Hosts that frequently exchange a high volume of packets with many other nodes will naturally receive a higher score, indicating their strategic importance and potential risk if compromised. This design ensures that the results can be easily interpreted and integrated with the other modules of the risk evaluation model.

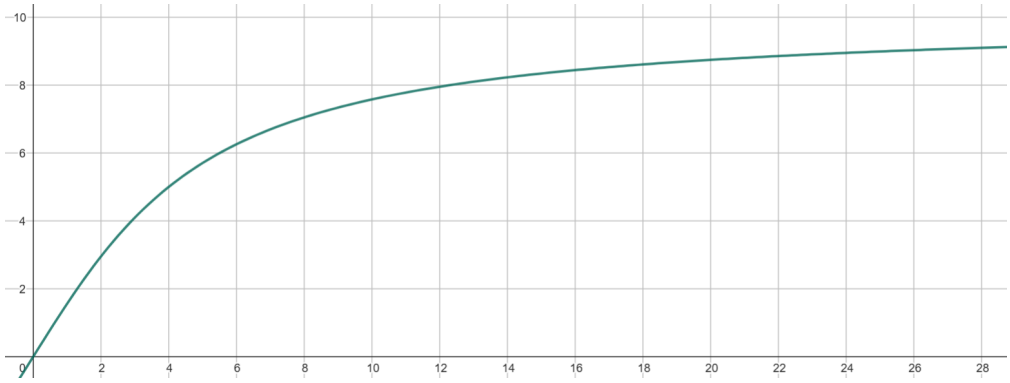

Figure 4.6: Graphical representation of the topology module scoring formula

4.5 Description of the asset appraisal module

The asset appraisal module represents a fundamental component of the proposed risk evaluation framework, as it focuses on assessing the strategic, operational, and business relevance of each individual asset within the network. While other modules concentrate on more technical aspects, such as vulnerability exploitation, topological centrality, or attack propagation, this module shifts the perspective toward the consequences of a potential compromise.

It aims to evaluate the criticality of an asset by analyzing various contextual dimensions, including the services hosted, the type of data it processes or stores, the dependencies other systems have on it, and its role within essential business workflows.

By integrating this perspective into the overall model, the asset appraisal module supports risk-based prioritization, helping stakeholders distinguish between assets that are merely vulnerable and those that are both vulnerable and mission-critical. This enables the model to provide recommendations that are not only technically sound but also aligned with the organization's strategic goals and risk tolerance.

4.5.1 Formula Definition

Unlike other modules in the model that rely on precise mathematical expressions or automated data collection to compute a risk score, the asset appraisal module adopts a more qualitative and human-centric approach.

It is based on the use of a structured questionnaire that aims to capture critical information not readily accessible through automated scanning tools or passive observation of the network. This includes elements such as the business function of each asset, the type and sensitivity of the data it processes or stores, its availability requirements, its role in supporting key business operations, and its exposure to external threats. These aspects are vital to understanding the real impact that the compromise or failure of an asset could have on the organization, yet they often require human judgment or input from business stakeholders, system administrators, or security officers.

The questionnaire is designed to be comprehensive and scalable, incorporating weighted questions and conditional logic to adapt to different types of systems or organizational contexts. The answers provided are aggregated to produce a final numerical score that can be compared with the outputs of other modules in the model. This ensures consistency across the overall risk evaluation process while allowing the system to account for factors that go beyond technical configurations or network behavior.

With the inclusion of human insight into the equation, the asset appraisal module ensures that critical assets are accurately prioritized, even if their technical footprint appears modest. This blended approach, combining subjective assessment with structured scoring, contributes significantly to a more realistic and business-aligned understanding of cyber risk.

The questionnaire used in the asset appraisal module is designed with a dynamic structure, where the path of the evaluation can vary depending on the answers provided. Certain responses trigger follow-up questions or activate specific sections of the questionnaire, ensuring that the assessment is both context-aware and tailored to the characteristics of each asset.

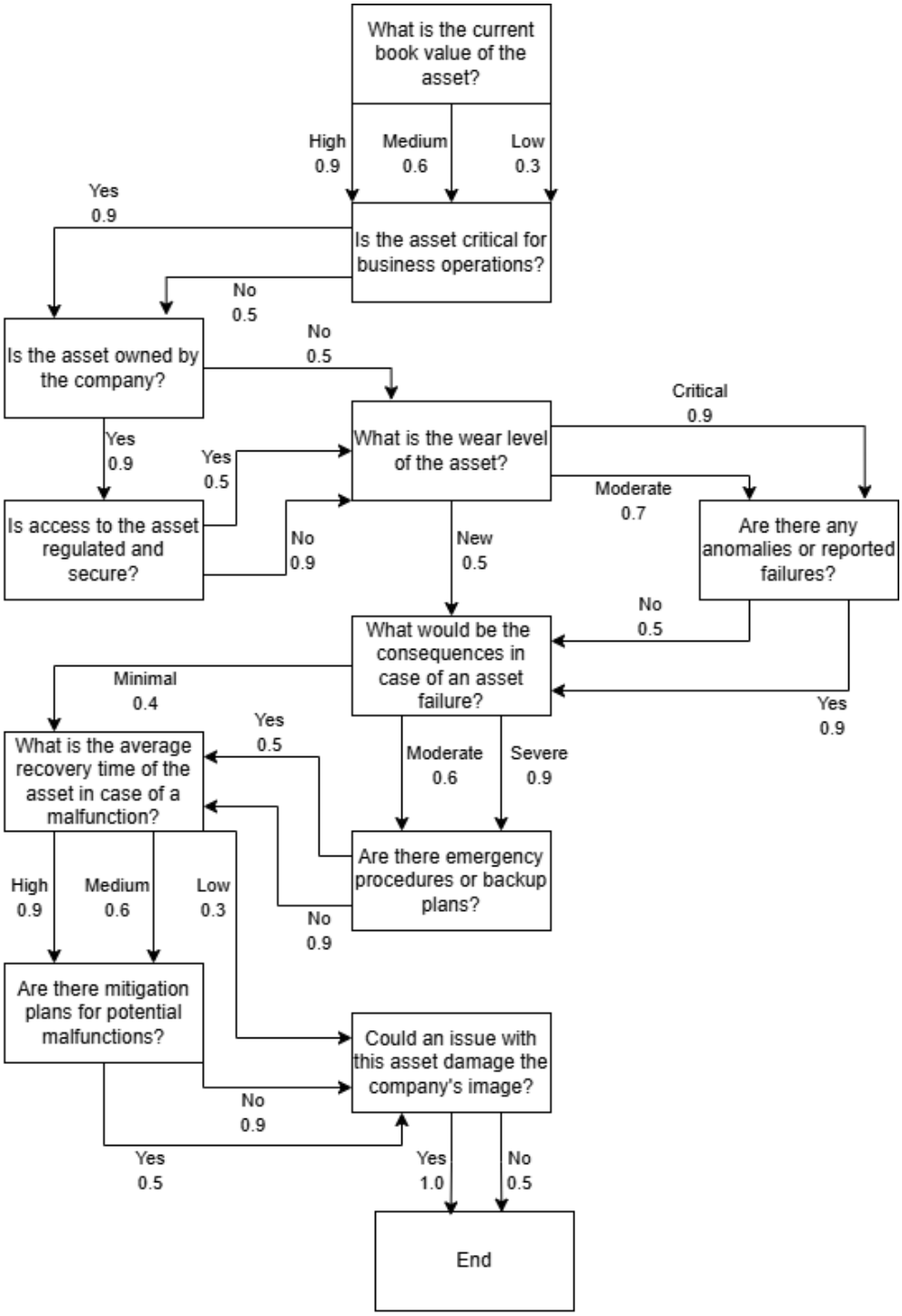

Figure 4.7: Visual representation of the asset appraisal questionnaire

This conditional branching mechanism allows for a more detailed investigation of critical systems while avoiding unnecessary questions for less significant components. Each question and corresponding answer is assigned a predefined weight based on its relevance to the asset's overall importance and risk exposure. At the end of the evaluation, all collected weights are summed to produce a final score, representing the asset's criticality.

4.5.2 Definition of input data

The input to the asset appraisal module is provided through the manual completion of this structured questionnaire by the network administrator or another individual with comprehensive knowledge of the infrastructure. This process ensures that all the contextual and operational details, which are not accessible through automated means, are accurately captured and integrated into the assessment. Since these inputs involve strategic and organizational insights, such as the business relevance of a host or its role within critical operations, they require human judgment and cannot be inferred solely from technical scans.

The questionnaire serves as a guided tool to facilitate this input, helping the administrator provide consistent and meaningful data for each asset under evaluation. However, in large-scale networks, evaluating every single host individually would be inefficient and time-consuming. To address this challenge, the module includes the option to group similar assets (those with comparable roles, configurations, and risk profiles) into logical clusters. This aggregation strategy makes the input process more scalable while still retaining the fidelity needed for effective risk modeling.

This approach streamlines the input process while still preserving the integrity of the data collected, ensuring that the risk evaluation remains comprehensive without overwhelming the user.

4.5.3 Definition of output data

The output of the asset appraisal module is a numerical score, normalized within the range [0, 10], that reflects the overall criticality and exposure of each evaluated asset. This value is obtained by aggregating the weighted responses from the questionnaire, which follow different paths based on the input provided by the administrator. Each answer contributes a specific score depending on its associated risk implication, and the final result is computed as the sum of these contributions. This structure allows the output to capture multiple dimensions of asset relevance, such as business importance, physical condition, security measures, and operational resilience. The modular nature of the questionnaire ensures that even partial or aggregated evaluations can still produce a meaningful output, making the system adaptable to networks of various sizes and complexity.

The final score is not only easy to interpret but is also directly compatible with the scoring outputs of the other modules.

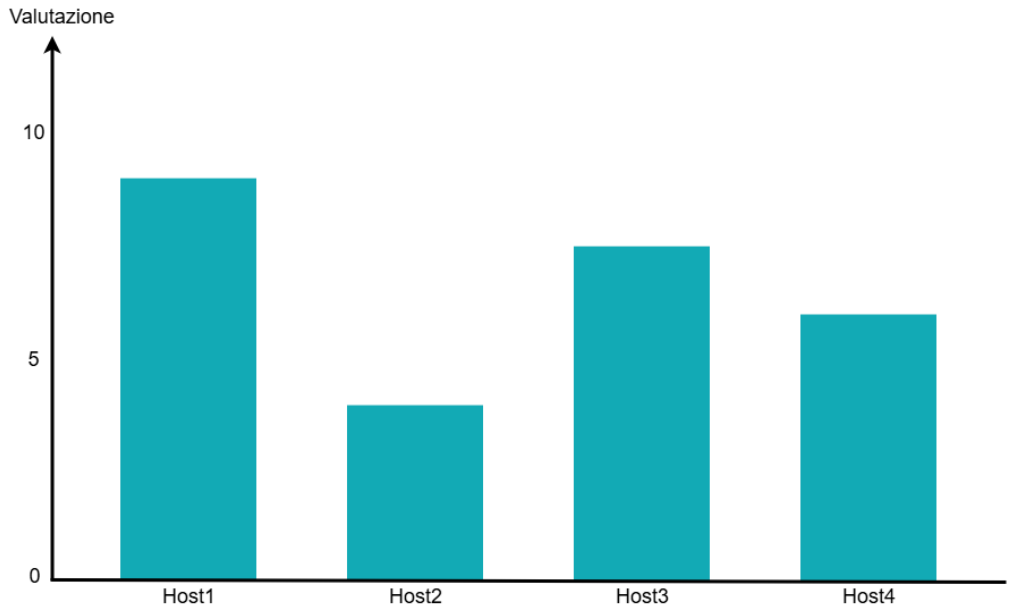

Figure 4.8: Graphical representation of the asset appraisal module scoring

4.6 The aggregator module

In complex environments where numerous systems and devices coexist, each with unique characteristics and levels of criticality, giving an overall score is important because it provides a synthetic and objective assessment of a system's security posture. By aggregating different factors, it produces a clear indicator that helps prioritize mitigation efforts. Without a comprehensive score, it would be more difficult to compare different systems and determine where to focus security efforts. The aggregator module is specifically designed to unify the previous evaluations into one.

The formula adopted by the aggregator module consists of computing the arithmetic mean of the scores generated by the individual evaluation modules. Each of the previous scores captures a different facet of an asset's overall exposure to cyber threats, from technical weaknesses to its structural role in the network and its business value.

Through averaging these scores, the module provides a perspective on the risk level related to each host.

The final score is normalized within the range of 0-10, maintaining consistency in all evaluated systems.

Although this method does not apply weights to individual components, the uniform contribution of each module ensures a balanced view that reflects the overall security posture without favoring any specific dimension.

4.7 Modules Comparison

Below is a summary table that describes the key components of the risk assessment model.

The table includes a brief description of each module and the formula used to compute its respective score, providing a comprehensive overview of how different risk factors are integrated into the final assessment.

Module	**Description**	**Formula**
Vulnerability	Evaluates the risk associated with known vulnerabilities based on exploitability and usage in attacks.	$\frac{20}{\pi} \cdot \arctan(\sum(\text{CVSS}^{1.5} \cdot \text{EPSS} \cdot A_s))$
Attack Graph	Assesses the criticality of each host within the attack graph structure.	$\frac{20}{\pi} \cdot \arctan\left(\frac{x}{\frac{N_{ag}}{N_{total}}}\right)$, where $x = \sum(\text{parents} + \text{children}) \cdot \text{modifier}$
Topology	Scores the centrality of a host based on communications and traffic volume.	$\frac{20}{\pi} \cdot \arctan\left(\frac{nc \cdot ph}{nh}\right)$
Asset Appraisal	Measures business value and criticality through a structured questionnaire.	Sum of weighted answers from adaptive questionnaire
Aggregator	Combines the scores from all modules into a single final value.	Arithmetic mean of the individual module scores

Table 4.3: Summary table of the modules in the risk assessment model

Chapter 5

Results and evaluation

This chapter presents and analyzes the key results obtained from validating the engine's performance using a test network configuration.

Due to time related constraints, and to the focus of the work being mainly on research and model design; the engine has been tested on three different manually constructed network configurations, in order to provide a simple proof of concept of the correct functioning of the proposed solution.

In this chapter, the test network configurations will be presented along with the corresponding results produced by the risk assessment model. Each network will be described briefly to highlight its structure and specific characteristics, followed by an analysis of the outputs generated by the engine. The data model reported in Section A.1 will provide an example of the input data used by the model.

5.1 First test network configuration

As shown in the image, the test case scenario adopted in this work reflects a typical OT network infrastructure, characterized by the presence of devices commonly used in this technological domain (Section 2.1.2). The network includes key operational components such as *PLCs*, *HMIs*, a *SCADA* control server, and various I/O devices that represent sensors and actuators used for monitoring and controlling physical processes. The network architecture, on the other hand, also corresponds to that of a typical OT architecture, being *SCADA* (Section 2.1.3), as suggested by the presence of the *SCADA*.

5.1.1 Vulnerabilities

In order to perform the analysis a set of vulnerabilities has been generated for this network configuration.

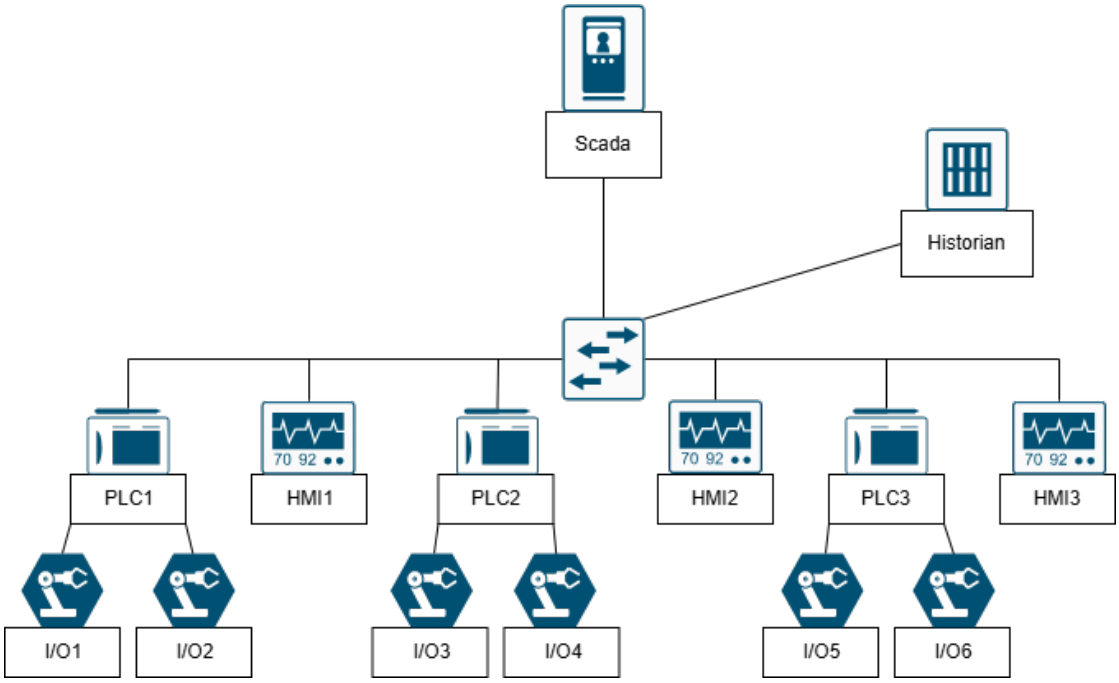

Figure 5.1: First test network configuration

A total of six distinct attack paths involving vulnerabilities were identified, targeting various components within the architecture. The HMIs (*HMI1*, *HMI2*, and *HMI3*) are particularly exposed, with a shared vulnerability scoring almost the maximum severity with a CVSS score of 9.8 and an EPSS of 0.8, indicating both the critical impact and the high likelihood of exploitation. *PLC1*, *PLC2*, *PLC3* are affected by additional vulnerabilities, with severity scores of 7.5 and EPSS of 0.51 that generate a set of attacks between 2 and 5. A last vulnerability has been inserted on the *SCADAPC* a CVSS score of 7.2.

In summary, the vulnerabilities span across both control and interface components, emphasizing the importance of comprehensive security across the entire OT stack.

5.1.2 Attack graph

An attack graph was generated based on the identified vulnerabilities and the manually defined network topology, in order to visualize the possible attack paths and assess how threats could propagate through the system.

It presents initial nodes involving *PLC1* and *SCADA*PC that act as entry points, with no predecessors, indicating they may be directly exploitable due to publicly accessible services or exposed vulnerabilities. From these points, attacks

can propagate through different hosts before converging into a central node, from which start a wide array of target nodes, playing a key role as a pivot point in the network. It symbolizes a critical stage in the attack progression, one where an attacker has successfully compromised the *SCADA* server and can now leverage this access to target nearly all other assets in the network. This kind of centrality not only highlights the strategic importance of this host but also reflects how its compromise could lead to a cascade of further intrusions.

The structure of the graph shows a highly interconnected environment where multiple attack paths converge toward critical assets like *PLCs* and the *SCADA* server, emphasizing the importance of a layered defense strategy. The inclusion of shared nodes, such as *Historian* or *HMIs*, also reveals how certain hosts can contribute to the spread of attacks across different segments.

5.1.3 Topology

A possible communication configuration was manually crafted ad hoc to simulate a realistic industrial environment.

SCADAPC stands out as a highly interconnected node, exchanging a significant number of packets with almost every other device in the network reflects its function as the central supervisory and control unit, which aligns with its role in typical *SCADA* architectures described in 2.1.3. The *PLCs*, on the other hand, show more targeted communication patterns. Each PLC primarily communicates with the *SCADAPC*, one or more *HMIs*, and the *Historian*, which is consistent with their dedicated role in managing specific sections of the industrial process. In particular, *PLC3* appears to exchange a relatively high number of packets with *SCADA*PC and the *Historian*, suggesting a more complex or critical process. The *Historian* functions as a data aggregation point, communicating with multiple nodes, including *SCADA*PC and the *PLCs*. This behavior fits its typical function of storing historical data collected across the network.

Overall, the topology exhibits a partial star structure centered around *SCADAPC*.

5.1.4 Asset appraisal

A possible evaluation of the assets from a business and economic perspective should take into account the role and criticality of each device within the industrial process.

Assets such as the *SCADA* has a central role in maintaining operational continuity, its failure or compromise could lead to production downtime, financial loss, and safety hazards. The *PLCs*, although individually less complex, control specific machinery or physical processes making their reliability essential to maintaining the efficiency and continuity of operations. For *HMIs*, while their compromise may

not immediately stop processes, it can lead to incorrect decision making, misconfigurations, or unauthorized actions that compromise the integrity and efficiency of the system. The *Historian*, tasked with collecting and archiving operational data, is crucial from a compliance, audit, and troubleshooting standpoint. Its value lies in enabling forensic analysis and supporting optimization efforts.

5.1.5 Analysis of obtained results

Taking these input data into account, the model generated the following results:

Host	Vulnerability	Attack Graph	Topology	Asset Appraisal	Final score
SCADAPC	7.1	9.5	9.8	9.6	8.5
HMI1	9.5	8.9	8.4	6.4	8.3
HMI2	9.5	8.5	8.4	6.4	8.2
HMI3	9.5	8.7	8.0	6.4	8.2
PLC1	8.8	9.4	9.2	7.1	8.5
PLC2	8.2	7.9	9.2	7.1	8.1
PLC3	9.1	7.9	9.3	7.1	8.3
Historian	0.0	6.8	9.5	8.7	6.2

Table 5.1: Final scores for the first network configuration

The *SCADAPC* and *PLC1* stand out with consistently high scores across all evaluation criteria, confirming their central role in the infrastructure and the associated criticality in terms of security. The *HMI* devices, while having high vulnerability scores, show slightly lower importance in the asset appraisal, which brings their final score just below the central nodes. The *PLC2* and *PLC3* maintain strong performance with a good balance of low vulnerabilities and high topological importance. On the other hand, the *Historian* shows a drastically low vulnerability score (likely due to the absence of detected issues), but retains relatively good scores in other areas. Nonetheless, its overall final score is the lowest among the hosts, suggesting a lower immediate priority in risk mitigation—though not to be overlooked.

5.2 Second test network configuration

The network configuration shown in Figure 5.2 represents a more complex and hierarchical topology compared to the previous one. Although it maintains the same fundamental components, *PLCs*, *HMIs*, *SCADA* PC, *Historian*, it introduces a clearer segmentation between different zones. Additional components such as an operator *workstation* and a laptop are included. The separation of functions

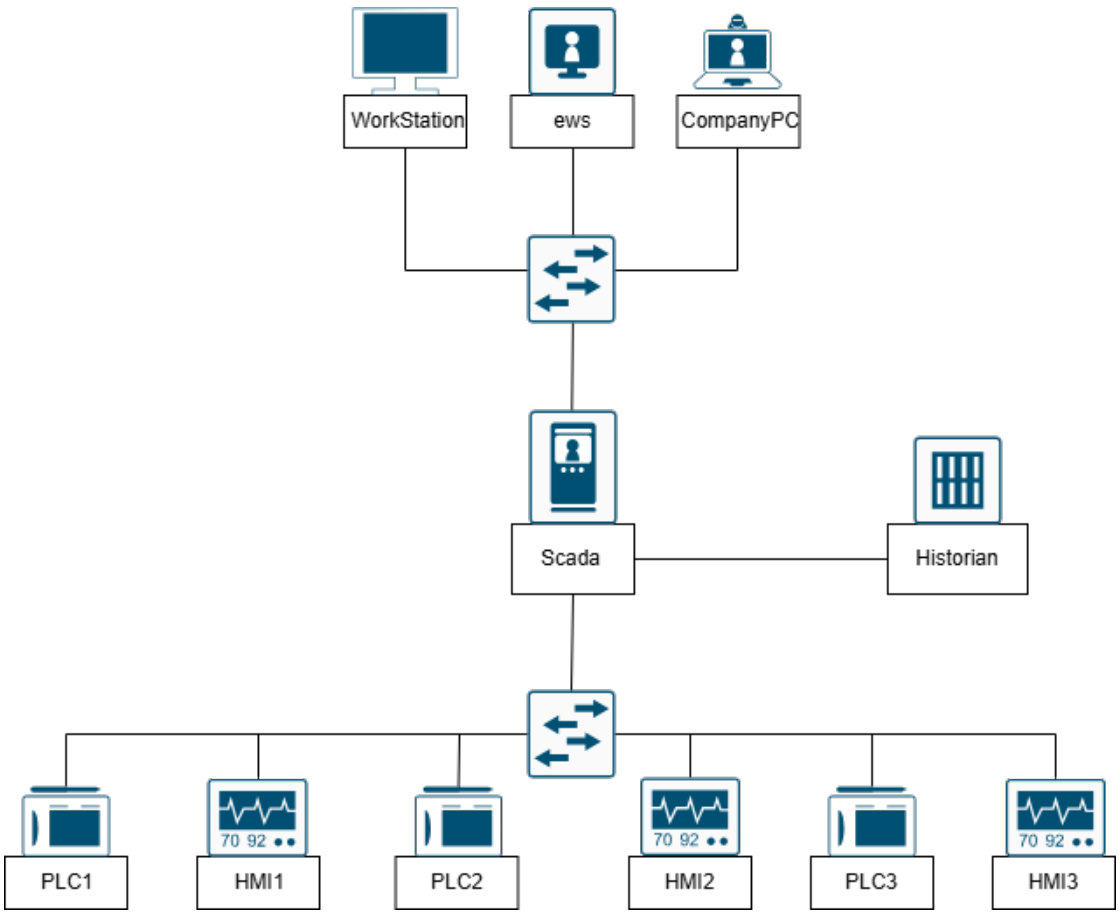

Figure 5.2: Second test network configuration

and enhanced control flow reflect a more realistic and structured OT environment, making it closer to what would be expected in a real-world industrial setting.

5.2.1 Vulnerabilities

The vulnerability data set for the second network configuration reveals a total of 8 attacks distributed on several critical devices, highlighting multiple potential entry points for adversaries.

HMIs exhibit the most severe vulnerability, with a CVSS score of 9.5 and a high EPSS percentile of 0.78, indicating both a high impact and a high probability of exploitation. The *SCADA* PC also presents a notable risk, with a score of 7.8 and being the target of 4 attacks, the highest in the dataset, making it a central concern for network defense. The *PLCs* each show medium to high vulnerabilities (scores between 6.9 and 7.5) and are collectively targeted by 8 attacks, confirming

their exposure and importance in the control infrastructure. Finally, the *Historian*, although less targeted, still has a moderate vulnerability, emphasizing that even support systems should not be overlooked.

5.2.2 Attack graph

The attack graph associated with the second network configuration outlines a structured and multistage exploitation path.

The attack sequence initiates from *PLC1*, which acts as a gateway to two paths: one leading to the *Historian* and the other to a joint state involving both *PLC1* and the *Historian*. These paths converge toward the *SCADA* PC, marking it as a strategic pivot point for further attack propagation. From the *SCADA* PC, attackers gain access to the *HMIs*, which in turn open multiple paths toward high-value targets including *PLC3* and ultimately a compromised state involving both the *SCADA* PC and *PLC3*. The inclusion of *EWS*, accessed via both *SCADA* PC and HMI nodes, underlines the risk to administrative *workstation*s and their role in facilitating lateral movement.

The graph emphasizes how exploitation of initial low-level assets such as *PLCs* and the *Historian* can escalate to control over central systems like the *SCADA* PC and *HMIs*.

5.2.3 Topology

Compared to the previous topology, this network exhibits higher traffic density and a more intricate structure, reflecting a mature industrial setup with robust data flows.

The *SCADAPC* emerges as the central node, showing high-volume, bidirectional traffic with nearly every other host, including significant exchanges with *PLC1*, the *Historian*, and *HMI1* underscoring its critical function in centralized control and coordination. The *PLCs* maintain frequent communication with both *SCADA* and the *HMIs*, reflecting their operational role in real-time data acquisition and control. *PLC1* shows the most intense traffic levels with *SCADA* and *HMI1*, indicating its strategic placement within the process chain. The *Historian* server is highly active as well, particularly with *SCADA* and the *PLCs*, consistent with its role in aggregating and archiving process data. Meanwhile, the *CompanyPC*, though less active, maintains consistent communication with core systems like *SCADA* and *EWS*, suggesting a supervisory or business-side interface. The *HMI* devices are strongly connected to both *SCADA* and the *PLCs*, affirming their function as the primary human interface points in the control system.

5.2.4 Asset appraisal

From a business perspective, the *SCADAPC* is the most critical asset as it represents the core of operations and any disruption could result in significant production downtime and financial loss. The *PLCs* are also highly rated, since they directly impact production continuity and process reliability: key factors for maintaining business output. The *EWS* supports essential functions like system configuration and recovery, which are vital to minimize operational delays. The *Historian* plays a relevant role in storing process data, compliance, and performance optimization, supporting long-term business planning. The *HMIs* are important for daily operations but less impactful if lost, as redundancy can mitigate short-term effects. Lastly, the Company PC have minimal influence on production processes and business continuity, serving mostly as a support asset.

5.2.5 Analysis of obtained results

Considering these input data, the model produced the following outcomes:

Host	Vulnerability	Attack Graph	Topology	Asset Appraisal	Final score
workST	0.0	0.0	7.4	5.5	3.2
CompanyPC	0.0	0.0	7.1	5.1	3.0
PLC1	8.5	8.6	9.7	9.2	9.0
PLC2	7.4	5.6	8.6	8.9	7.6
PLC3	8.4	9.0	9.3	8.9	8.9
HMI1	8.9	8.4	8.5	7.1	8.2
HMI2	8.9	8.4	7.9	6.6	7.9
HMI3	8.9	8.4	7.9	6.6	7.9
SCADAPC	7.3	9.4	9.0	9.6	8.8
Historian	4.4	9.2	8.4	7.6	7.4
EWS	0.0	7.5	6.6	8.3	5.6

Table 5.2: Final scores for the second network configuration

PLC1 and *PLC3* emerge as the most critical assets indicating both high vulnerability and strategic importance in the network. *SCADA* PC also ranks highly due to its central role in control operations and business reliance. The *HMIs* score between 7.95 and 8.23, showing that they are significant targets with notable vulnerabilities and functional importance. *PLC2* and the *Historian* also hold considerable weight with scores above 7. The *EWS* and user devices such as *CompanyPC* and *workstation* score significantly lower, reflecting either limited exposure to attacks or lower overall importance from an operational perspective.

5.3 Third test network configuration

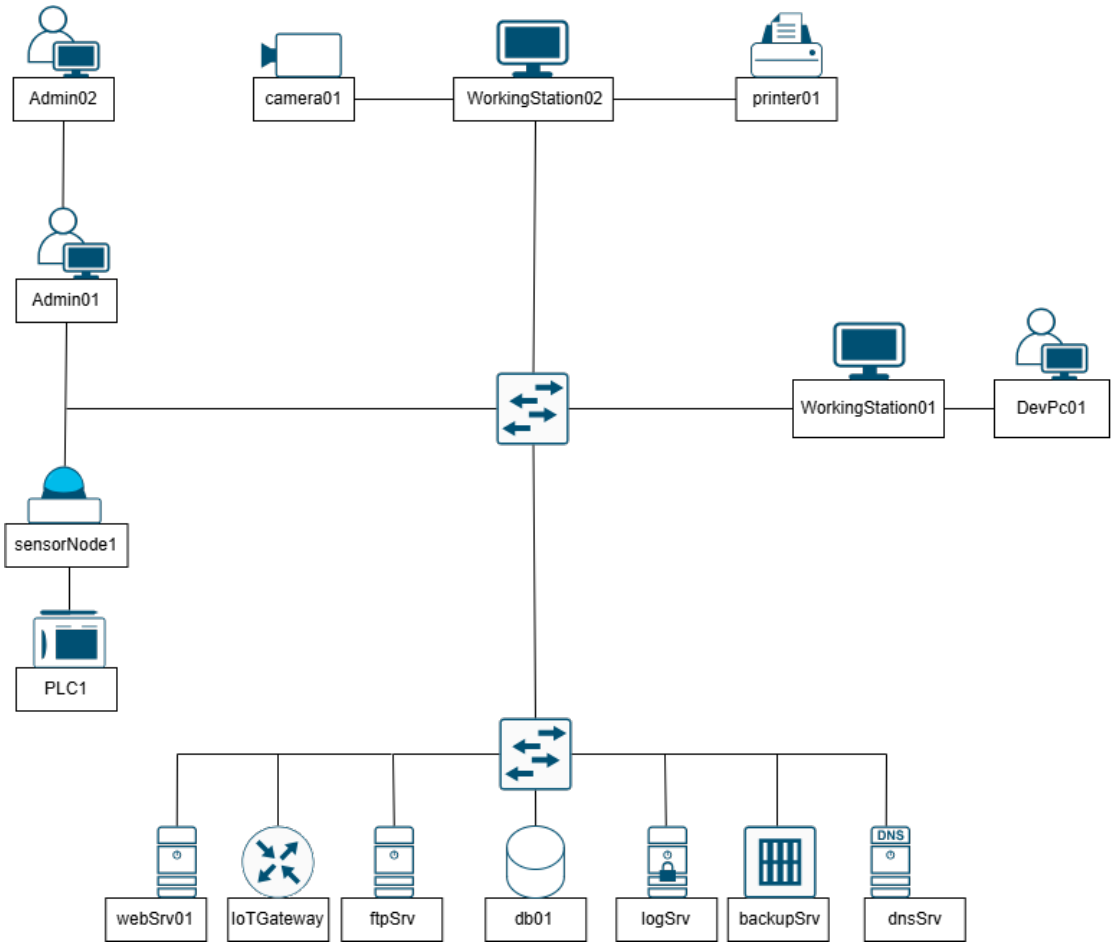

Figure 5.3: Third test network configuration

This network represents a more segmented and service oriented architecture compared to the previous one. It includes a clear division between administrative nodes, user workstations and critical infrastructure. It also features a back-end segment with essential services interconnected via a centralized switching layer. Compared to the earlier network, this topology appears more structured and layered, with better isolation between control systems and support services.

It shows a higher level of maturity in terms of network design, enabling more granular control over traffic flow and potentially improved security management.

5.3.1 Vulnerabilities

The vulnerability dataset for this network architecture reveals a total of 10 detected attacks distributed across several critical assets.

The most heavily targeted hosts include *PLC1*, *ftpSrv*, *webSrv01*, and *db01*, all of which exhibit high vulnerability scores and are each associated with three attack attempts. The *ftpSrv* is particularly concerning, with the highest CVSS score of 8.0 and an EPSS value of 0.70, indicating both a severe impact and a high probability of exploitation. Similarly, *PLC1* shows a vulnerability score of 7.8 and an EPSS of 0.53, making it a high-priority target due to its critical operational role. *webSrv01* and *db01* also present substantial risks, with vulnerability scores of 7.4 and 7.6 respectively, and multiple recorded attacks. These assets are often exposed to external access or serve as central data hubs, which increases the attack surface and potential damage from successful exploitation. Other hosts such as *workstation*01, *dnsSrv*, and *logSrv* exhibit lower scores and fewer attacks, suggesting a lower risk profile.

Overall, the distribution of vulnerabilities and attacks suggests a mix of opportunistic and targeted activity, with attackers focusing on systems that are both exposed and critical to operations.

5.3.2 Attack graph

The initial entry points appear to be *PLC1*, *sensorNode1*, *Admin01*, *workstation01*, and *devPC01*, all of which lack parent nodes, suggesting that they are exposed or accessible entry points for attackers. From *PLC1*, the attack can pivot to *iotGateway*, which serves as a crucial intermediary, further branching into *db01* and *ftpSrv*. *db01*, once compromised, provides access to *logSrv*, which then connects to *webSrv01*, a central node in the graph due to its multiple incoming and outgoing connections. *webSrv01* plays a pivotal position, receiving input from both *logSrv* and *workstation01*, and passing potential attacks to *dnsSrv* and ultimately to *logSrv* and backupSrv, which marks the endpoint of some attack paths. The presence of multiple converging paths into *webSrv01* and *dnsSrv* emphasizes their importance as potential chokepoints or escalation vectors. Other nodes like *Admin01* facilitate indirect access to *ftpSrv* and, eventually, *dnsSrv*, indicating multiple viable lateral movement strategies for an attacker. Peripheral nodes such as *camera01* and *printer01* appear to have limited involvement, with *camera01* contributing indirectly via *iotGateway*, while *printer01* remains isolated.

Overall, this graph highlights how a compromise of key nodes like *iotGateway*, *webSrv01*, or *ftpSrv* can enable broad access to critical infrastructure components.

5.3.3 Topology

The topology has been designed as a moderately dense and functionally segmented network, where administrative and core service nodes demonstrate the highest interaction, while control, peripheral, and end-user systems form distinct communication clusters.

The most active nodes are *Admin02* and *Admin01* engaging in substantial data exchanges with multiple devices suggesting a strong administrative or monitoring role. Similarly, *db01* shows high volumes both sent and received with *Admin01* and with *ftpSrv*, confirming its central role in data management and its high interactivity with core services. Control-related devices like *PLC1*, *iotGateway*, and *sensorNode1* exhibit moderate to high packet exchanges with specific nodes, reflecting tightly bound interactions within the control subsystem. End-user devices such as *workstation01* and *workstation02* also display relevant communication patterns, particularly with *logSrv* and *webSrv01*, suggesting active user engagement and logging. Peripheral hosts like *camera01*, *printer01*, and *devPC01* show limited traffic, indicating their auxiliary function with low security impact in terms of connectivity. The presence of multiple moderate to high volume routes placed the focus on potential routes for lateral movement in the event of compromise.

5.3.4 Asset appraisal

The database server (*db01*) receives the highest score, as it stores critical business data essential for daily operations and long-term strategy. Similarly, the log server (*logSrv*) and *Admins* score very high due to their vital roles in security monitoring and access control. The *PLC1*, which directly governs industrial processes is rated just below, reflecting its key roles in maintaining production and system configuration. The backup server follows closely with, being crucial for recovery in case of failures. Assets like the *iotGateway*, web server, and DNS server have a average score, signifying their supporting roles in communication and system access. General purpose devices such as workstations and developer PCs are moderately important, while peripheral or single function devices like sensor nodes, printers, and cameras receive lower ratings due to their limited impact on core business functionality.

5.3.5 Analysis of obtained results

Based on the provided input data, the model generated the following results:

Host	Vulnerability	Attack Graph	Topology	Asset Appraisal	Final score
Admin02	0.0	0.0	7.5	8.9	4.1
Admin01	0.0	6.0	6.1	8.4	5.1
PLC1	8.2	7.8	4.4	8.4	7.2
sensorNode1	0.0	6.0	2.5	5.7	3.5
iotGateway	6.2	8.2	3.5	7.5	6.3
workstation01	2.6	6.0	5.0	6.2	4.9
workstation02	0.0	0.0	3.8	6.0	2.4
printer01	0.0	0.0	2.2	3.2	1.3
webSrv01	8.3	8.6	2.1	7.1	6.5
ftpSrv	8.1	7.4	3.8	6.6	6.5
db01	7.6	7.4	9.7	9.7	8.6
logSrv	1.7	9.0	7.6	8.9	6.8
backupSrv	0.0	7.8	4.0	8.1	5.0
camera01	0.0	6.0	2.3	3.2	2.9
dnsSrv	2.4	8.2	3.5	7.5	5.4
devPC01	0.0	6.0	2.1	6.2	3.6

Table 5.3: Final scores for the third network configuration

The highest score is assigned to *db01*, reflecting its critical role in storing valuable business data and its high exposure in terms of both vulnerabilities and network connectivity. Hosts like *webSrv01* and *ftpSrv* also score high, due to their significant external exposure and presence of severe vulnerabilities, making them essential components from a risk and operational perspective. *LogSrv* and *iotGateway* follow closely, underlining their importance in terms of data flow monitoring and interconnectivity. On the lower end, devices like *printer01*, *camera01*, and *workstation02* show minimal strategic relevance and limited vulnerability, resulting in low overall scores. Intermediate scores are seen for *Admin01*, *PLC1*, and *dnsSrv*, reflecting moderate risk and business importance.

Overall, the scores indicate that in this configuration, data storage, communication, and external-facing services are the most business-critical assets.

5.4 Validation

In conclusion, the results obtained across the various network configurations provide a structured assessment of risk and asset importance within different industrial and corporate environments. Through the integration of vulnerability scores, attack graph analysis, topological relevance, and asset criticality, the model produces comprehensive evaluations that support decision-making in cybersecurity management.

However, the validation of these outcomes presents significant challenges. Due to the scarcity of comparable methodologies and the limited availability of reference

datasets or benchmarking studies in the current literature, it is difficult to quantitatively verify the model's accuracy. This limitation underlines the complexity of evaluating cybersecurity risk in diverse and evolving network architectures.

To address this issue and strengthen the reliability of the findings, a peer review process is planned. The results will be submitted to industry experts and academic professionals in the field of cybersecurity for evaluation and feedback. Their insights will be crucial in refining the model, validating the approach, and ensuring the robustness and practical applicability of the proposed methodology.

Chapter 6

Conclusions and future work

This work aims to address the increasing concerns surrounding cybersecurity in OT infrastructures by proposing an automated solution that supports the security assessment process, leveraging evaluation techniques to enhance accuracy and reliability.

It is important to note, however, that the prototype engine presented in this work is not intended to function as a standalone solution. Instead, it has been specifically designed for seamless integration within the broader company project, serving as a component of the automated security assessment pipeline. In support of this integration, a standardized input model for representing network data has also been defined.

Within the context of the project, the main objective of this work is the development of a reporting engine capable of automatically generating comprehensive security reports. These reports summarize the evaluation results of the target system by integrating data such as network topology, identified vulnerabilities, asset relevance, and possible attack paths.

In order to achieve its prescribed goal, the presented prototype engine uses a set of different evaluation frameworks as the core part of the project. It then proceed to build upon them by adding data from different sources, including network topology, known vulnerabilities, attack graph analysis, and asset appraisal. This multilayered approach enables the generation of a thorough and reliable overview of the network's current security posture. The final output is structured in a way that not only identifies critical weaknesses but also facilitates risk prioritization and guides mitigation planning, making it a valuable tool for decision makers and security analysts alike.

As showcased in Chapter 5, the engine prototype is capable of correctly assign risk score to the different hosts. However, this has only been tested for a small number of cases of network configurations that have been manually constructed to include diverse situations. Going forward, it would therefore be of great benefit to

add support for more types of input, aiming at covering real world scenarios.

Also the modules of the model could definitely undergo some improvement, such as enriching the asset appraisal by integrating detailed inventory data and operational dependencies, and improving the comparability aspect across different scenarios of the formulas.

It may be beneficial to draw inspiration from [35] and explore the use of deep reinforcement learning algorithms as a means to derive optimal scoring strategies.

Overall, the solution presented in this work marks a solid first step toward the creation of an automated tool for the security assessment of OT networks, with the potential to evolve into a standard reference within the industrial landscape.

Bibliography

[1] Keith Stouffer, Michael Pease, C Tang, Timothy Zimmerman, Victoria Pillitteri, and Suzanne Lightman. «Guide to operational technology (ot) security». In: *National Institute of Standards and Technology: Gaithersburg, MD, USA* (2022) (cit. on pp. 6, 8, 15, 16, 18, 19, 22).

[2] Mohammad Omar Abdullah Ephrem Ryan Alphonsus. «A review on the applications of programmable logic controllers (PLCs)». In: *Science Direct* (2016) (cit. on p. 9).

[3] Michael Tiegelkamp and Karl-Heinz John. *IEC 61131-3: Programming industrial automation systems*. Vol. 166. Springer, 2010 (cit. on p. 9).

[4] Francis Enejo Idachaba. «Review of Remote Terminal Unit (RTU) and Gateways for Digital Oilfield delpoyments». In: *(IJACSA) International Journal of Advanced Computer Science and Applications* (2012) (cit. on p. 9).

[5] John Angelopoulos Dimitris Mourtzis and Nikos Panopoulos. «The Future of the Human–Machine Interface (HMI) in Society 5.0». In: *MDPI* (2023) (cit. on p. 10).

[6] Martin Gergeleit Alexios Karagiozidis. «A Forensic Analysis Framework for Industrial Control Systems». In: *Association for Computing Machinery* (2025) (cit. on p. 11).

[7] Bitton Ron et al. «Deriving a Cost-Effective Digital Twin of an ICS to Facilitate Security Evaluation». In: *Springer, Cham* (2018) (cit. on p. 11).

[8] Geeta Yadav and Kolin Paul. «Architecture and security of SCADA systems: A review». In: *International Journal of Critical Infrastructure Protection* 34 (2021), p. 100433. ISSN: 1874-5482. DOI: https://doi.org/10.1016/j.ijcip.2021.100433. URL: https://www.sciencedirect.com/science/article/pii/S1874548221000251 (cit. on p. 14).

[9] Wolfgang Mahnke, Stefan-Helmut Leitner, and Matthias Damm. *OPC unified architecture*. Springer Science & Business Media, 2009 (cit. on p. 20).

[10] George Thomas. «Introduction to the modbus protocol». In: *The Extension* 9.4 (2008), pp. 1–4 (cit. on p. 20).

[11] Igor Belai and Peter Drahoš. «The industrial communication systems Profibus and PROFInet». In: *Applied Natural Sciences* 1 (2009), pp. 329–336 (cit. on p. 21).

[12] Frank Cremer. «Cyber risk and cybersecurity: a systematic review of data availability». In: (2022) (cit. on p. 25).

[13] Elvis Agbadoku. «Cyber Risk Management: The Impact of Data in the Assessment of Cyber Risk by Cyber Insurers». In: *Reasearch Gate* (2024) (cit. on p. 30).

[14] *Common Vulnerability Scoring System v3.0: Specification Document*. FIRST (cit. on p. 31).

[15] Assane Gueye and Peter Mell. *A Historical and Statistical Studyof the Software Vulnerability Landscape*. 2021. arXiv: 2102.01722 [cs.CR]. URL: https://arxiv.org/abs/2102.01722 (cit. on p. 33).

[16] Fabio Massacci Luca Allodi. «Security Events and Vulnerability Data for Cybersecurity Risk Estimation». In: (2017) (cit. on p. 33).

[17] Peter Mell Karen Scarfone. *The Common Configuration Scoring System (CCSS): Metrics for Software Security Configuration Vulnerabilities*. NIST (cit. on p. 34).

[18] Karen Scarfone. «Vulnerability scoring for security configuration settings». In: *ACM* (2008) (cit. on p. 34).

[19] Peter Mell Elizabeth LeMay Karen Scarfone. *The Common Misuse Scoring System (CMSS): Metrics for Software Feature Misuse Vulnerabilities*. NIST (cit. on p. 35).

[20] Daniel Mellado, Eduardo Fernández-Medina, and Mario Piattini. «A comparison of software design security metrics». In: *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*. ECSA '10. Copenhagen, Denmark: Association for Computing Machinery, 2010, pp. 236–242. ISBN: 9781450301794. DOI: 10.1145/1842752.1842797. URL: https://doi.org/10.1145/1842752.1842797 (cit. on p. 35).

[21] Fabio Massacci Luca Allodi. «Evaluating Exploit Prediction Scoring Systems: A Systematic Analysis of EPSS». In: (2023) (cit. on p. 36).

[22] Rianna Parla. *Efficacy of EPSS in High Severity CVEs found in KEV*. 2024. arXiv: 2411.02618 [cs.CR]. URL: https://arxiv.org/abs/2411.02618 (cit. on p. 37).

[23] Peter Mell Elizabeth LeMay Karen Scarfone. *Guide to Industrial Control Systems (ICS) Security*. NIST (cit. on p. 37).

[24] Stephen Quinn David Waltermire. *The Technical Specification for the Security Content Automation Protocol (SCAP)*. NIST (cit. on p. 37).

[25] *NIST Interagency Report 8286C*. NIST (cit. on p. 38).

[26] Anass Zaidouni, Mohammed Abdou Janati Idrissi, and Adil Bellabdaoui. «A PRISMA Systematic Review for Intelligent IS/IT Project Portfolio Dashboard from the Value, Prioritization and Risk Perspectives». In: *2024 4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*. 2024, pp. 1–14. DOI: 10.1109/IRASET60544.2024.10548997 (cit. on p. 39).

[27] *Engineering Trustworthy Secure Systems*. NIST (cit. on p. 39).

[28] Ugur Saritac, Xingya Liu, and Ruhai Wang. «Assessment of Cybersecurity Framework in Critical Infrastructures». In: *2022 IEEE Delhi Section Conference (DELCON)*. 2022, pp. 1–4. DOI: 10.1109/DELCON54057.2022.9753250 (cit. on p. 41).

[29] Jonathan M. Spring et al. *Prioritizing Vulnerability Response: A Stakeholder-Specific Vulnerability Categorization*. Carnegie Mellon University (cit. on p. 41).

[30] Mikiko Kikuchi. «What is SSVC, a new metric for security vulnerability assessment?» In: *NTT DATA Group Corporation* (2024) (cit. on p. 42).

[31] Joaquin Garcia-Alfaro Marwan Lazrag Christophe Kiennert. «Quantifying the Impact Propagation of Cyber Attacks using Business Logic Modeling». In: *Springer* (2024) (cit. on p. 44).

[32] Albina Orlando. «Cyber Risk Quantification: Investigating the Role ofCyber Value at Risk». In: *MDPI* (2021) (cit. on p. 45).

[33] Ioannis Agrafiotis Arnau Erola. «A system to calculate Cyber Value-at-Risk». In: *Science Direct* (2022) (cit. on p. 45).

[34] Giulio Sunder, Alberto Salvatore Colletto, Sara Raimondi, Cataldo Basile, Alessio Viticchié, and Alessandro Aliberti. «Enhancing OT Threat Modelling: An Effective Rule-Based Approach for Attack Graph Generation». In: *ICSC: Intelligent Cybersecurity Conference*. 2024 (cit. on p. 54).

[35] Zhenguo Hu, Razvan Beuran, and Yasuo Tan. «Automated Penetration Testing Using Deep Reinforcement Learning». In: *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. 2020, pp. 2–10. DOI: 10.1109/EuroSPW51379.2020.00010 (cit. on p. 79).

Appendix A

Supplementary material

This appendix offers supplementary material referenced throughout the dissertation. It contains key data models employed during the project's development phase, along with brief descriptions. The essential features of these models are further discussed in detail within the specific chapters that reference them.

A.1 JSON model of the tested network environment

This section provides the JSON data structure modelling one of the network scenarios for which the proposed tool has been tested.

It is important to note that, for the purpose of this work, the following file was created manually, as at the time of development, the module of the tool responsible for performing automatic network reconnaissance was under construction. However, in the feature this model is expected to be generated automatically.

The network configuration that this file refers to is described in Section 5.2, and corresponds to the second test case.


```
{
    "hosts": [
      { "ID": 1, "name": "workST" },
      { "ID": 2, "name": "companyPC" },
      { "ID": 3, "name": "plc1" },
      { "ID": 4, "name": "plc2" },
      { "ID": 5, "name": "plc3" },
      { "ID": 6, "name": "hmi1" },
      { "ID": 7, "name": "hmi2" },
      { "ID": 8, "name": "hmi3" },
      { "ID": 9, "name": "scadaPC" },
```

```
      { "ID": 10, "name": "historian" },
      { "ID": 11, "name": "ews" }
    ]
  }
```

A.1.1 JSON model for the vulnerabilities

This section provides an example of the JSON data structure used to model the vulnerabilities present in the system as discussed in Section 4.2.

```
{
  "tot_attacks": 8,
  "vulnerabilities": [
      {
          "ID": 1,
          "host": ["plc1"],
          "score": 7.5,
          "EPSS": 0.55,
          "attacks": 3
      },
      {
          "ID": 2,
          "host": ["plc2"],
          "score": 6.9,
          "EPSS": 0.50,
          "attacks": 2
      },
      {
          "ID": 3,
          "host": ["plc3"],
          "score": 7.3,
          "EPSS": 0.52,
          "attacks": 3
      },
      {
          "ID": 4,
          "host": ["hmi1", "hmi2", "hmi3"],
          "score": 9.5,
          "EPSS": 0.78,
          "attacks": 2
      },
      {
          "ID": 5,
          "host": ["scadaPC"],
          "score": 7.8,
```

```
            "EPSS": 0.20,
            "attacks": 4
        },
        {
            "ID": 6,
            "host": ["historian"],
            "score": 6.5,
            "EPSS": 0.40,
            "attacks": 1
        }
    ]
}
```


A.1.2 JSON model for the attack graph

This section presents a sample JSON data structure used to represent the attack graph of the system, as discussed in Section 4.3.


```
{
  "nodes": [
    {
        "ID": 1,
        "host": ["plc1"],
        "fathers": [],
        "sons": [2, 3]
    },
    {
        "ID": 2,
        "host": ["historian"],
        "fathers": [1],
        "sons": [6]
    },
    {
        "ID": 3,
        "host": ["plc1", "historian"],
        "fathers": [1],
        "sons": [4]
    },
    {
        "ID": 4,
        "host": ["scadaPC"],
        "fathers": [3],
        "sons": [6, 7]
    },
    {
```

```
        "ID": 5,
        "host": ["plc2", "historian"],
        "fathers": [],
        "sons": [6]
    },
    {
        "ID": 6,
        "host": ["scadaPC"],
        "fathers": [2, 4, 5],
        "sons": [8]
    },
    {
        "ID": 7,
        "host": ["hmi1", "hmi2", "hmi3"],
        "fathers": [4],
        "sons": [8, 9, 10]
    },
    {
        "ID": 8,
        "host": ["scadaPC", "ews"],
        "fathers": [6, 7],
        "sons": []
    },
    {
        "ID": 9,
        "host": ["plc3"],
        "fathers": [7],
        "sons": [10]
    },
    {
        "ID": 10,
        "host": ["plc3", "historian"],
        "fathers": [7, 9],
        "sons": [11]
    },
    {
        "ID": 11,
        "host": ["scadaPC", "plc3"],
        "fathers": [10],
        "sons": []
    }
  ]
}
```

A.1.3 Data model used for the topology

The following table presents the traffic matrix representing the number of packets exchanged between each pair of hosts within the network. Each cell in the matrix is expressed as a pair of values in the format sent/received, where the first number indicates the packets sent from the row host to the column host, and the second number indicates the packets received in return.

	ScadaPC	**Historian**	**ews**	**workST**	**PLC1**	**PLC2**	**PLC3**	**HMI1**	**HMI2**	**HMI3**	**CompanyPC**
ScadaPC	0/0	30/20	21/12	15/25	45/78	23/58	35/44	24/36	23/31	17/21	28/34
Historian	20/30	0/0	12/6	14/5	36/52	28/38	33/46	25/57	21/36	24/35	26/29
ews	12/21	15/12	0/0	19/16	22/30	26/29	27/31	25/26	21/22	23/28	19/25
workST	25/15	22/21	22/22	0/0	24/34	27/37	29/33	23/25	24/26	25/30	20/21
PLC1	45/78	26/34	25/22	21/19	0/0	31/40	33/35	32/41	34/33	30/30	29/28
PLC2	58/23	29/32	27/25	26/23	32/29	0/0	30/38	27/32	29/31	26/25	24/26
PLC3	35/44	33/46	27/31	29/33	33/35	30/38	0/0	25/29	24/28	32/40	22/24
HMI1	24/36	25/57	25/26	23/25	32/41	27/32	25/29	0/0	28/30	26/28	21/22
HMI2	23/31	21/36	21/22	24/26	34/33	29/31	24/28	28/30	0/0	27/29	23/23
HMI3	17/21	24/35	23/28	25/30	30/30	26/25	32/40	26/28	27/29	0/0	22/20
CompanyPC	28/34	26/29	19/25	20/21	29/28	24/26	22/24	21/22	23/23	22/20	0/0

Table A.1: Traffic matrix between hosts (packets sent/received)