UNIVERSITY OF TRENTO

Department of Information Engineering and Computer Science

Bachelor's Degree in
Computer Science

Final Dissertation

An overview of ICS vulnerabilities, threats, and security measures

Based on three real-world power plants security concerns

Supervisor
Silvio Ranise

Student
Riccardo Gennaro

Academic Year 2021/2022

Acknowledgements

To express my gratitude to all the people that shaped my being during all of these years would require another thesis. But still, I will try and do that.

Firstly, I would like to thank my family, my father Michele and my mother Francesca in particular, who always stood by my side, giving me all their love and precious advice.

Secondly, I thank my friends and peers, both from Trento and Valdagno. In particular, Dr. Matteo Franzil and Dr. Claudio Facchinetti for standing as exemplary professional role models, and also Enrico Fornasa, friend, peer, and roommate who somehow was able to deal with me for nine years.

Thirdly, I would like to thank everyone who helped me develop this work. Prof. Silvio Ranise for being the supervisor of this thesis. Dr. Fabrizio de Cataldo for standing as a leadership model. Also, all my colleagues at Atos shaped my staying in Milan into an unprecedented experience.

Furthermore, I thank all the teachers that took part in molding my way of thinking and studying, Prof. Marco Zoso, Prof. Francesca Cocco, Prof. Anneliese Defranceschi, and Prof. Alberta Mistè for their undying passion for their fields of study.

Finally,

this work is dedicated to my grandfather, Valentino Danzo,
my grandmothers, Genoveffa Maltauro, and Feliciana Crosara,
and the memory of my grandfather Angelo Gennaro.

Contents

Abstract

The concept of *Industrial Control System* (ICS) has heavily changed during the last decades, but only during the last few years, we are seeing a change in security awareness surrounding these systems. As seen during recent and past events, and since this type of technology is also implemented in critical infrastructure, a failure in these environments can lead to severe consequences for society, the economy, and the environment.

Once *air-gapped*, i.e. not exposed to unsecured networks, ICSs were not subject to Internet threats. Nowadays, this architecture choice has been discarded in favor of remote command & control, remote emission monitoring, and data analysis. This caused these networks to be exposed to a plethora of threats. Since this paradigm shift was commonly not coupled with a high-security awareness, to this day a great number of industrial processes suffer from numerous security issues ranging from poor asset visibility to the use of unencrypted communication.

This work is divided into two parts. The first part presents an overview of security frameworks, *Supervisory Control And Data Acquisition* (SCADA) architecture, attack vectors, and adversary common *modus operandi*.

The second part consists of three case studies based on data gathered from the ICSs of three power plants. The studies were produced during an internship collaboration with Atos between March and April 2022 and continuously updated and revised during the following months. The project aimed at enhancing the security posture of nine different power plants, by applying a non-intrusive methodology.

1 Introduction

Since the start of the COVID-19 pandemic, three different trends have emerged and started to change the way people view industrial cybersecurity [22].

The need for an **increase in process efficiency**[2] causes companies to feel the urge to digitize their operations. This thread implies the need for an interconnected production floor, especially around remote operations. As a result, the standard industrial environments, once air-gapped from the IT apparatus, are today exposed to insecure networks, e.g. the Internet, and their threats.

Moreover, a **growth in cybercrime**[17][2] has been observed: only a few years ago, OT infrastructures were mainly targeted by nation-state attackers - the only actors with sufficient resources to target such systems - using complex and expensive-to-develop malicious software, such as the infamous Stuxnet worm. Nowadays, those sophisticated tools are leveraged by cybercriminal organizations and cause substantial damage (e.g. WannaCry ransomware).

Finally, recently we are witnessing **legislation and regulations tightening**. Aiming to protect sectors such as energy, utilities, and transportation - critical to both economy and national security - different legislators around the world are implementing new regulations while updating existing ones.

To adapt critical infrastructures to the legislation and overcome the main security issues, different solutions are or are becoming, available on the market, with Nozomi, Otorio, and Fortinet as the most relevant technologies. In the context of an industrial apparatus, being provided with a resilient, proactive, scalable, passive, and non-intrusive system is key to providing security and reliability to the whole internal network. Violating any of the CIA triad principles on those infrastructures would cause disastrous consequences not only to the attacked company but may impact national security (e.g. compromising a power plant).

This dissertation aims to study the inherent problems affecting those *Industrial Control Systems* (ICSs) that are at the core of all those infrastructures that are vital to our society.

To do so, this work is divided into two parts: the first will explore the architectures and frameworks employed in the design of an ICS, other than the risks and threats that they face; the second part offers three case studies of three different real-world power plants. Each of those is exposed to both the difficulty of gaining asset visibility and the security problems that can be caused by faulty design and poor maintenance.

Four chapters are presented. Chapter 1 offers an introduction to the fundamental concepts, terminology, and trends concerning ICSs security. Chapter 2 deepens what has been introduced in chapter 1 by describing the main implemented architectures and standards, and by presenting the main threats to, and security problems of an ICS. Chapter 3 presents the methodology used to better the asset visibility and to produce a vulnerability assessment. Chapter 4 describes the application of the methodology to the three, previously mentioned, networks. Finally, the conclusion is given, underlying the studies' results and calling for an increase in awareness.

1.1 Differences between IT and OT security

Informational Technology (IT) consists of all the software and systems, that enable people and machine communication and information exchange. On the other hand, *Operational Technology* (OT) uses hardware and software to manage and control industrial equipment and systems to produce physical changes during an industrial process.

Cybersecurity problems and solutions for IT and OT systems are fundamentally different[13]. While IT security protects devices and environments using standard solutions suitable for the more popular general-purpose OSs, OT security is challenged by specific-purpose OSs, proprietary communication protocols and software, and a lack of traditional security tools. Furthermore, there are different priorities: OT cybersecurity aims to provide service availability and worker safety, rather

than the confidentiality typically achieved by IT security.

Also, IT and OT face dissimilar types of security events and consequences[30]. A successful attack against an ICS could bring destructive outcomes, such as vast financial losses, critical infrastructure compromise, or even the loss of lives. Moreover, it is extremely difficult to patch or update an OT system compared to an IT system. This is caused by both the necessity to halt the production, and by the restrictive compatibility requirements that a SCADA system can require. As a consequence, the vast majority of ICSs run on unpatched software and obsolete devices.

Finally, operational environments require high responsiveness and fast data flow; as a consequence, all the deployed security solutions mustn't impact negatively on the system's performance[25].

1.2 Architectures and Standards

Industrial Automation and Control Systems (IACSs), or simply *Industrial Control Systems* (ICSs), are a type of control system commonly deployed in an industrial context to achieve assets safeness, and production efficiency and consistency.

Nowadays, *Supervisory Control And Data Acquisition* (SCADA) systems are the fundamental modules in an IACS architecture. Essentially, SCADA systems are utilized for plant and process monitoring, machine control, data collection and analysis, alarm monitoring, and reporting. All these functions are handled by different devices, i.e. PLCs, RTUs, HMIs, Databases, and so on. In Chapter 2 there will be additional information on these systems.

To build a secure and functioning IACS, different architectures and frameworks have been proposed, with the main ones being.

- **PERA**. Developed by professor Theodore Joseph Williams throughout the 1990s, the *Purdue Enterprise Reference Architecture* (PERA), also known as the Purdue model, is a well-known and widely adopted architecture for ICS security. This model aims to implement an air gap between the operational technology (OT) and the informational technology (IT) systems. To do so, the model offers a layered view of the IT/OT system comprising six layers ranging from 0 to 5, other than a DMZ[25].

- **IEC 62443**. Initially focused on industrial automation, and then shifted to cybersecurity design for *Industrial Automation and Control Systems* (IACSs), the IEC 62443 security framework consists of a series of standards, well known for providing "a methodology for applying security in operational and field environments for cyber-physical systems"[10].

- **The ICS Cyber Kill Chain**. This work is presented by its authors as a way to "help defenders understand the adversary's cyber attack campaign"[3]. It offers a representation of the attacker's point of view during the entirety of the attack campaign and it is based on two major attacks: *Stuxnet* and *Havex*.

Chapter 2 offers a more in-depth description of these topics, other than a high-level classification of the used attack vectors dividing them into hardware, software, and communication attacks.

2 Related work

This chapter is meant to give an overview of the architectures, standards, frameworks, and common threats related to the Operational Technologies.

2.1 SCADA Systems Architectures

Since the first deployments of SCADA infrastructures at the end of the 1950s, four main architectures have been delineated: monolithic, distributed, networked, and web-based. Also, a new generation of systems has arisen: IoT SCADA[20][1].

Since the first deployments of SCADA infrastructures at the end of the 1950s, five generations have been delineated.

Based on mainframes, the first type of architecture, called **monolithic** SCADA, initially used mini-computers boarding 8 to 16-bit processors. With the development of transistors and microprocessors, and the subsequent increase in CPU speed and memory size, real-time and reliable SCADA systems became a reality in the late 1960s. This implementation offered scarce scalability and standardization as, for the time being, there were no standardized communication protocols between masters and slaves; therefore, only proprietary protocols were used.

With the demand from the asset owners for a more scalable and reliable service and the introduction of LAN technology, different new types of distributed stations were included in SCADA systems. This new **distributed** infrastructure deployed *communication processors*, *Human-Computer Interfaces* (HMI), *Remote Terminal Units* (RTUs) and calculation processors.

As "the systems [...] were (still) limited to hardware, software and peripheral devices provided by the vendor"[1], and thanks to the growing number of automated supply chains and ICS vendors, the necessity for an open-system and standardized architecture became apparent.

The **networked** architecture is very similar to the distributed one, with the main difference being the open system approach: instead of being based on a proprietary environment, it uses standardized protocols such as IP. This generation of SCADA systems was not only able to communicate in a WAN, but also to integrate third-party hardware and software.

This rapid evolution called for standardization: in 1996, the first standard was developed - Object Linking and Embedding (OLE) for Process Control, specifying an interface for HMI/SCADA to interface to third-party devices - and the OPC (Open Platform for Communications) foundation was established[14].

Thanks to the growing capabilities of web applications, and the necessity for a reactive industrial control, remote access to the SCADA system was made available. This new **web-based** architecture is based on open-system protocols. The data flow, consisting of read-write commands, takes place on the Internet, and the Human-Machine Interfaces (HMIs) are accessible from generic browsers (e.g. Firefox, Google Chrome)[20].

Finally, **Internet of Things** (IoT) SCADA systems, are the latests frontier in term of industrial control technologies. This generation of systems makes extensive use of Industrial IoT devices in order to obtain an higher level of scalability, interoperability, and integration, other than enabling the SCADA system to better implement a variety of technologies such as edge computing, cloud computing, big data analysis, and machine-to-machine communication[29]. Of course, the deployment of Internet of Things devices on the OT network introduces new challenges to the security of these systems.

2.1.1 SCADA System Components

SCADA systems have a solid architecture containing several devices (stations) with different roles. Commonly, these components are: RTUs, PLCs, HMIs, Telemetry Systems, Data Acquisition Servers, Historians, and Supervisory Services[1].

- **Remote Terminal Units**. A *Remote Terminal Unit* (RTU) is a device acting as a data acquisition and control unit. This station is used to control processes and machinery to a remote location, and also to gather data from the ground units, converting it to digital data, and transfer it to the supervisory system via a telemetry system. Furthermore, nowadays RTUs are provided with a network interface, enabling the system integrator to set the process and machinery control with ease[5][1].

- **Programmable Logic Controller**. A *Programmable Logic Controller* (PLC) is a computer used in industrial control with similar roles to the RTUs ones. The main difference consists in the fact that a PLC needs to be programmed and can execute either simple or complex logic operations. The basic operation of this board are scanning the input, scanning the in-memory program, executing the program logic, updating the output to operate the devices, run self-diagnostics, and reporting to the master[5][1].

- **Telemetry Systems**. With telemetry systems, we refer to the set of communication protocols and physical channels used for inter-station communications[1]. The once-used proprietary protocols and telephone wires have been substituted with common standardized technologies such as TCP, IP, Ethernet 802.3, and Wi-FI 802.11.[5][29].

- **Human-Machine Interface**. The *Human-Machine Interface* (HMI) consists of the *Graphical User Interface* of the SCADA system and it's an aid to the human operators that allows them to interact with the ICS[1]. Commonly, the HMI is a piece of software deployed on a general-purpose OS - e.g. Windows, or Linux.

- **Data Acquisition Server**. This component offers an interface for data acquisition between the PLCs and RTUs, and the Historian. It's based on a client-server architecture and industrial protocols[1].

- **Historian**. This part of the architecture is used to store the data generated by the PLCs and the RTUs, such as alerts, events, process variables' values, timestamps, and so on. Furthermore, it is possible to run queries against this database to generate reports that can be displayed on the various HMIs[1].

- **Supervisory Control**. This module is the core of a SCADA system. *Supervisory control* is usually a computer running software containing the logic and algorithms that govern all of the automatized processes. It is capable to read and sending commands to all the RTUs and PLCs in order to handle events and alerts[1].

2.2 Frameworks and standards

2.2.1 The Purdue Model

As previously explained in section 1.2, PERA is a vastly implemented architecture for Industrial Control Systems, ad it is divided into six levels[25].

- **Level 5: Enterprise network**. The enterprise network serves the purpose of collecting data from the systems in the below levels and reporting different production metrics such as inventory, compliance with the production plan, and so on. Even though this level is not part of the ICS, it depends on the data produced from the latter to drive strategic decisions[25]. This layer also deploys a Demilitarized Zone between the enterprise network and the Internet. A *Demilitarized Zone* (DMZ) is a subnet that contains and offers a set of services to an insecure net, in this case, the Internet.

- **Level 4: Business planning and logistics**. This layer consists of all the corporate IT systems that support the operational processes. At level 4 we can find technologies such as *Enterprise Resource Planning* (ERP) systems, comprising of logistic, administrative, profits, losses, and production management, other than data-driven decision-making supports based on

data warehousing, data mining, and so on. Furthermore, this level houses web servers, email clients, etc[25][27].

- **Industrial Demilitarized Zone**. The Purdue model implements the DMZ via a double firewall architecture. Despite this intermediate layer being key for providing at least the most basic form of security to the OT infrastructure, in most organizations it is absent or has limited capabilities. Inside the DMZ we can find systems such as proxies, application servers, historian mirrors, and so on[25]. Furthermore, we are now witnessing a merge between the fourth and the third levels caused by the need for bidirectional data flows between the IT and OT systems. This phenomenon is known as OT/IT convergence[24][31].

- **Level 3: Site operations**. At this level are implemented all the systems with plant control and monitoring functions such as HMI engineering workstations and historians. This is the highest level belonging to the OT system, and it's the one that collects all the data generated by the lowest levels and reports to the fourth level (IT)[25].

- **Level 2: Supervisory Control**. Level 2 is a drill-down of the level 3 view. As in the level above, HMIs are present but are used to supervise and control smaller portions of the industrial system[25].

- **Level 1: Basic Control**. This level is dedicated to PLCs and RTUs, equipment capable of control controlling machinery as seen in section 2.1.1 [25].

- **Level 0: Physical Process**. The last layer describes all the processes, machinery, actuators, and sensors. The systems belonging to this level are also called *Equipment Under Control* (EUC)[25].

Following, figure2.1 gives a graphical representation of the model.

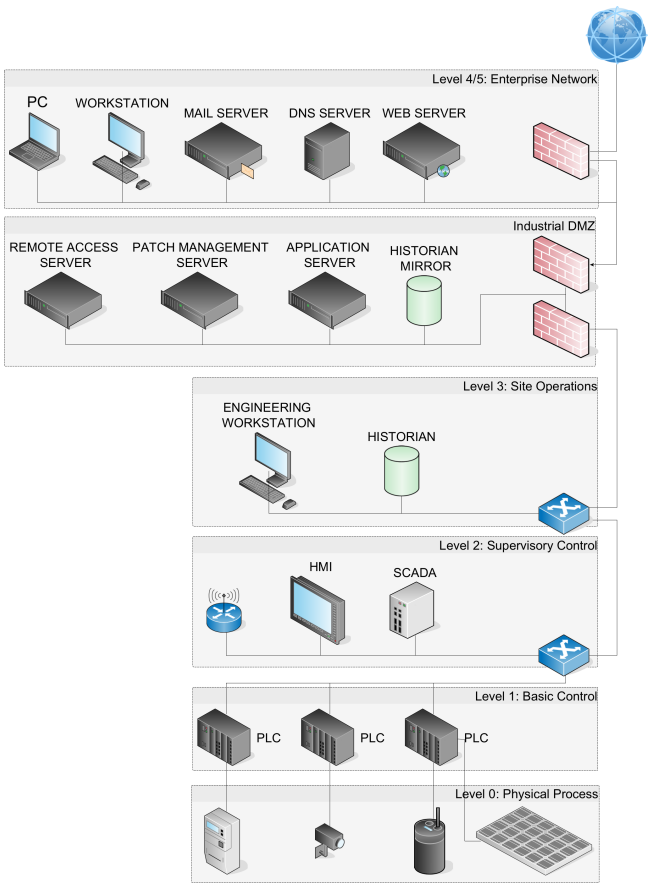

Figure 2.1: Purdue model.

2.2.2 ISA/IEC 62443 Framework

To improve the safety, reliability, integrity, and security of the ICSs, the *International Society of Automation* (ISA) developed the ISA/IEC 62443 family of standards. This family defines a set of common terms and requirements that the various actors that take part in the development of these systems should satisfy.

The structure of ISA/IEC 62443 is divided into four groups, as depicted in figure 2.2[19].

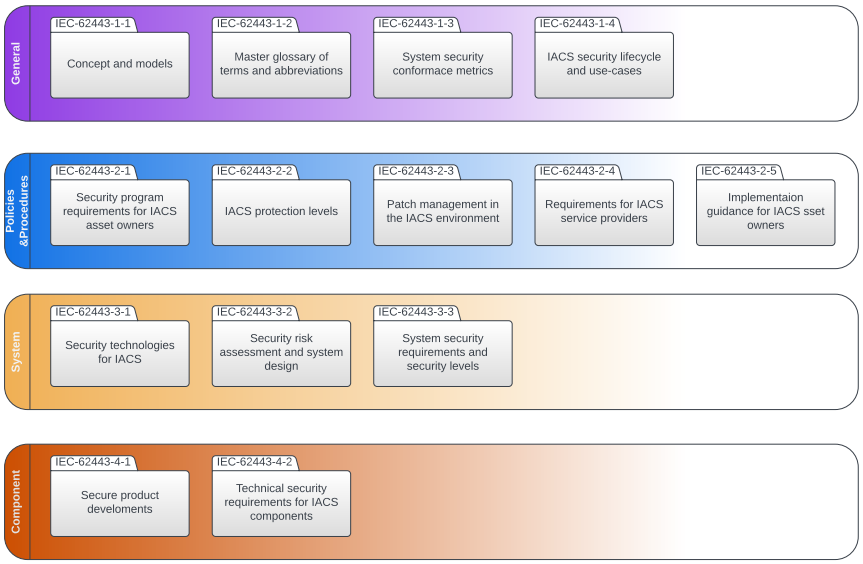

Figure 2.2: IEC 62443 structure.

Since the intended audience of this standard interests multiple actors, such as those who design, implement, manage, and operate the system, a discussion over all of the documents would be cumbersome. For this reason, we constrict the discussion to the fundamental concepts related to the security of the automation network.

Differently from the CIA triad employed in traditional information security, this framework introduces a variation of this triad based on Safety, Reliability, and Availability[26]. Furthermore, the foundational requirements of ISA/IEC 62443 for the ICSs offer a deeper view of the security requirements than the one presented in the CIA triad.

The aforementioned requirements are[26]:

- **Access Control**. Role-based access control is needed to guarantee that only authorized users access the resources. Also, the principle of least privilege must be satisfied.
- **Use Control**. The usage history of the devices must be logged to protect the system against persistent threats, and also in order to facilitate any future forensics.
- **Data Integrity**. Not different from the concept of Integrity in the CIA triad.
- **Data Confidentiality**. Not different from the concept of Confidentiality in the CIA triad.
- **Restrict Communication**. The network must be properly segmented. Document 3-2 of the standard propose a segmentation based on "Zone and Conduits", where a zone is defined as a "grouping of logical or physical assets based upon risk or other criteria such as criticality of assets, operational function, physical or logical location, required access, or responsible organization"; while a conduit is defined as "a logical grouping of communication channels that share common security requirements connecting two or more zones"[19].
- **Timely Response to Events**. The response time is a strong theme in ICSs security. During the 2015 cyber attack on the Ukrainian power grid, the adversary put particular emphasis in slowing down the response time of the defenders. This was done by wiping the disk of those

Windows systems linked to serial-ethernet communications, i.e. the communication between the EWSs and the power breakers, in order to delay the restoration efforts[4]. For this reason, designing a protocol to restore or halt the system in case of hijack is mission critical.

- **Resource Availability**. In order to maintain a stable output of the end service, it is necessary to ensure the availability of the network.

To satisfy all of these requirements, document 3-2 of the standard offers an overview of the implementable security measures and technologies for an ICS.

The standard also presents a qualitative scale to represent the level of security of a zone. This score can help in the security management of the subnets. Following are the definitions of the four security levels[10].

- **SL 1**. Protection against casual or coincidental violation.
- **SL 2**. Protection against intentional violation using simple means with low resources, generic skills, and low motivation.
- **SL 3**. Protection against intentional violation using sophisticated means with moderate resources, ICS-specific skills, and moderate motivation.
- **SL 4**. Protection against intentional violation using sophisticated means with extended resources, ICS-specific skills, and high motivation.

Dr. Daniel DesRuisseaux's white paper for Schneider Electric [8], offer a more detailed description of these security levels.

2.3 The ICS Cyber Kill Chain

To design a secure ICS, a cybersecurity engineer must have a deep knowledge of both the underlying process and the adversary's attack methodologies. Here we explore the latter by reporting the key concepts of the *ICS Cyber Kill Chain* presented in 2015 by Michael J.Assante and Robert M. Lee[3].

Although this work is based on two relatively old case studies, Stuxnet and Havex, the presented attack methodology can still describe, at a high-level, more recent attacks like the Ukraine power grid hacks of December 2015[4] and December 2016[23].

The ICS Kill Chain describes an attack on an ICS in two stages: the first aims to exfiltrate information regarding the control system and to access production environments; the second aims to exploit the acquired knowledge to develop and tune a capability able to compromise the ICS to, successively, deliver and installing it.

2.3.1 Kill Chain First Stage

The first stage is composed by the actions presented below.

- **Reconnaissance**. This phase mainly consists of *passive and active reconnaissance*. The first consists of searching public information that can support the attack, namely information about employees, networks, processes, and protocols used. The latter aims at mapping the ICS attack surface and learning activity patterns.
- **Preparation**. During this stage, the adversary focus on *targeting* and *weaponization*. The attacker must identify a vector for his/her attack taking into consideration effort, the likelihood of success, and detection risk. Weaponization has intended the act of modifying a harmless file to take advantage of a native feature. In the case of the 2015 attack against the Kyiv power grid, MS Excel and Word documents were altered by embedding BlackEnergy 3 and delivered via a spearphishing campaign[4].
- **Cyber Intrusion**. The intrusion is divided into three following steps.

The first step, *Delivery*, consists of employing some techniques to interact with the targeted network; as per the previously mentioned attack, the delivery means several emails containing the weaponized file sent to the administration or IT departments[4].

The second step, *Exploit*, represents the means the attacker uses to tamper with the network, e.g. exploitation of a vulnerability upon the opening of a weaponized file, or the exploit of a compromised VPN.

The third step, *Install*, describes the installation of a piece of malicious software; in the case of the used example, upon opening the modified file and enabling document macros, the malware, BlackEnergy 3, was able to install[4].

- **Command & Control**. After the installment of the capability, the adversary can establish a persistent threat via the aforementioned malware or by an abused VPN connection. It is common practice to establish multiple Command & Control (C2) paths in order to maintain access to the network even if one path is severed down.

At this point, the attacker can be classified as a persistent actor. From now on the adversary will act to carry out his/her main goal. Common practices at this point consist of lateral movement, asset discovery, installation of secondary capabilities - e.g. KillDisk used with BlackEnergy3 during the previously cited 2015 cyber attack[4].

As seen in the attack used as an example, the capabilities installed during this stage are used to access the network or to slow down the recovery process, not to carry out the final goal. Nor BlackEnergy3, killDisk or the firmware modification of the PLCs was the cause of the energy outage. The cause of the outage was the remote manipulation of the engineering workstations used to control the station brakes.

2.3.2 Kill Chain Second Stage

During this stage, the attacker develops, test, deliver, install, and, finally, executes a capability that can satisfy the requirement of the planned attack.

- **Development**. In this step, the adversary uses the acquired knowledge about the ICS to develop an effective capability that will enable him/her to carry out the attack.

- **Testing**. Commonly performed on physical test appliances, the testing phase is performed to make sure that the capability can deliver a meaningful and reliable impact to the targeted network.

- **ICS Attack**. The last step consists of delivering, installing, and executing the designed capability. In our example, it would be a SCADA hijacking to open the station breakers[4].

In the set of meaningful malicious acts that can be carried out against an ICS, the various attacks are characterized by differences in complexity and aftermath.

For example, causing a DoS through TCP SYN flood attack is easier than manipulating sensors and actuators while also ensuring future replicability; but, on the other hand, the latter attack can cause significantly grater damage to the targeted organization.

2.4 Attack Vectors for ICSs

In the above presented Kill Chain is described as a common methodology that an adversary could follow to gain a foothold in the network. In this section, we use the SCADA architecture to model the different classes of actions that an adversary can take once he/she has access to the network. [16][18][30]

2.4.1 Attacks on hardware

Once an attacker gains access to a field device, he/she can manipulate the values of the variables generated or read by the device.

For example, during the 2009 Stuxnet attack, a hardware attack was mounted on the Siemens PLC controlling the motors of the centrifuge, in the end causing the explosion of the latters[12].

Not only hardware attacks can cause a device anomalous behavior, but they can also delay the defenders' response by feeding the engineering workstations tampered data, or by disabling alarms and other safety features. Also, this technique was used during the Stuxnet attack[12].

To mitigate this type of attack, robust access control must be implemented. Only by using strong credentials and logging access attempts an adversary action can be detected[30].

2.4.2 Attack on software

A SCADA system employs a massive variety of software to satisfy its requirements. Vulnerabilities can reside anywhere in Purdue architecture. In particular, there are two points of interest were a software attack can be mounted: historians and field device implementations.

The databases contained in a data *historian* can store valuable information for an attacker to use to complete the Development phase of the Kill Chain. This data includes, but is not limited to, past data used in information-driven algorithms or business decision processes, ICS technical data and specifications, etc. For this reason, ensuring that the queries launched against a database are properly sanitized is vital[30]. It is also important to take into consideration *False Data Injection Attacks* (FDIAs).

G. Liang *et al.*[21], offers a more in-depth review of the defense strategies against FDIAs and their impact on these systems.

Concerning the software mounted in *field devices*, it commonly presents various vulnerabilities caused by scarce privilege separation and memory buffer management.

This is caused by the fact that multiple embedded operating systems employed in SCADA systems, such as VxWorks[30], presented a monolithic kernel, with all of the applications running at kernel level. Also, multiple buffer overflow bugs are caused by bad software implementation; a common issue, since the majority of the code is written in C language.

2.4.3 Attack on communication

As in a common IT network, also ICS networks are exposed to the exploit of the implementation's vulnerabilities of their communication protocols. Instead of describing the classical attack launched against the TCP/IP stack, e.g. *SYN flood*, *DNS forgery*, *ARP poisoning*, this work should gloss over the vulnerability of common application-layer protocols used in SCADA systems.

Two of the most widely used communication protocols used in SCADA systems are presented below. The description of these two protocols is based on the conference proceeding *'Critical Infrastructure Protection III'*, in particular on I.N. Fovino et al. [15] paper and on S. East et al.[9] paper, that offers an in-depth attack taxonomy for these protocols.

Modbus protocol

Modbus is the *de facto* standard protocol for master/slave communication in industrial networks. There exists different implementations of this protocol; among those is also present Modbus TCP. This protocol presents two different types of communications, the first being *request-reply*, the second *broadcast*[15].

Modbus cannot be considered a secure communication protocol. Modbus does not satisfy the confidentiality and integrity requirements as no cryptography is implemented. Furthermore, no access control is implemented between master/slave communication. Finally, there is an absence of anti-repudiation or anti-replay mechanisms. It is important to keep in mind that this protocol, as DPN3, was designed to be lightweight as the devices that mounted, and mounted, their implementations have limited computational power.

Common attacks to this protocol are:

- **Unauthorized command execution**, caused by the absence of any authentication between master and slaves;
- **DoS attack**, triggered by a continuous and meaningless flux of data sent to a pool of slaves;

- **MitM attack**, caused by the unencrypted communication;
- **Replay attack**, triggered by sending a previously sniffed legitimate message.

Note that the majority of these vulnerabilities can be mitigated by operating some form of passive defense on the network, e.g. using IDS/IPS or by logging/monitoring network traffic and events.

DNP3 protocol

DNP3 is an application protocol widely used by electricity and water suppliers. Sitting directly above the TCP/IP or UDP/IP layers, this protocol is used to implement the propagation of SCADA system control commands and process data[5].

DNP3 supports three communication models. The first is based on *unicast* transactions between physically separated devices. This communication model is used for events management, e.g. a master requests (*reads*) the physical pressure to one of its slaves, or causes a valve to open (*writes*) via one of its slaves. The second consists of *broadcast* communication. The third model is *unsolicited responses*, typically used to provide periodical updates[9]. As for Modbus, also DPN3 does not employ encryption, authentication, or authorization, simply assuming that all the messages are valid.

Among the 28 attacks and 91 attack instances that S. East et al. [9] present in their work, three attacks are highlighted as the most common and the most threatening.

- **Passive Network Reconnaissance**. Given the lack of encryption, an adversary that has access to the network can easily intercept the traffic, store it and analyze it to gain information about the network structure and the employed devices.
- **Baseline Response Replay**. Once an attacker has gained knowledge about the communication patterns of the network, he/she can send forged responses and queries to simulate a master/slave variable exchange.
- **Denial-of-View**. In the case an adversary mounts a MitM device between a master and a pool of slaves, he/she is enabled to read, modify and forge the network traffic. Doing so will deny the defenders a truthful view of the network and the industrial process.

Differently from what has been seen for the Modbus protocol, simply analyzing the network traffic won't do the job. It is advisable to scan the network for duplicated IP and MAC addresses to detect any MitM attack attempt.

3 Methodology

This chapter describes the methodology used to produce a vulnerability assessment and mitigation for the threats to a third-party ICS. This research aims at exposing common issues in ICSs by bettering asset visibility without obstructing the network traffic. Since we are working on a third-party industrial control system, the main challenge is gathering information about the network and proposing solutions without having direct access to the infrastructure. To cause as less interference as possible to mission-critical communications, this methodology is based on a passive approach.

The content of this chapter describes a platform-independent solution.

3.1 Introduction

During the study of the network implementation, vulnerabilities and mitigation proposal, we assume that:

- there exists an *Operational* and *Enterprise Networks* where all of the communications are based on the Internet protocol suite;
- the enterprise and operational networks architecture follow the Purdue Model (cf. section 2.2.1), with the only difference being the absence of an *Industrial Demilitarized Zone*.
- a *SCADA system* with components as described in section 2.1.1 is present in the network. The architecture of this system match the *web-based* or *IoT* ones as in section 2.1;
- **the implemented solution has minimum impact on the operational network performance.** As the OT systems require a high refresh rate to control sensors and actuators, all security activities must be as passive as possible;
- **the end-service must remain available.** Since we are securing a critical infrastructure, in this case, a thermal power plant, all of the applied steps mustn't disrupt the end service.

The output of the methodology is an asset inventory and a set of mitigation.
This methodology follows the steps below:

- **Data gathering**: as a first step, data regarding the network schema and the current security posture is collected from the asset owner. As in most industrial systems, there is no real profound asset visibility, the second step is to do an inventory of all the network devices using passive automated asset discovery;
- **Vulnerability assessment**: once an asset inventory has been drowned down, we need to perform a vulnerability identification and monitor the network to outline possible threat scenarios;
- **Mitigation proposals**: when a set of threats, incidents, or alerts is identified, remediation and mitigation are proposed.

3.2 Data gathering

Before diving into the process of asset discovery, the network scheme is analyzed to define a choke point - i.e. a strategic location in the network where it is useful to analyze the data traffic, such as switches and firewalls - where an automated asset[1] discovery appliance can be deployed. To have the clearest view of the operational network, the choke point should be located between the third and second levels of the Purdue Model.

[1] An asset is a physical or virtual device used in the given network

Once the choke point(s) has been designated, a *Switched Port Analyzer* (SPAN) is set up and connected to a NIC (network interface card) placed in *promiscuous mode*[2] to mirror the network's traffic and perform a passive asset discovery. The mirrored data is not copied in its entirety, instead, we are interested in inferring:

- **the asset properties** being the IP and MAC address other than the type of appliance (e.g. PLC, HMI, etc), the operative system, and the firmware version.
- **the communication properties**, being the protocol used in the communication, the transferred bytes, the IP of the receiver, the date and time.

Unfortunately, a passive scan is not as reliable as an active one: devices that are not communicating won't be logged, and, not always, detailed information will be acquired, such as for operating systems' SP (Security Patch) numbers or firmware versions.

The result of this step is a database containing the assets in the network. In this database, an asset is a tuple in the form *(ID, name, type, os_or _firmware, IP, mac)*.

ID	Name	Type	OS/Firmware	IP	MAC
103b724d-d79d-453e-89d5	HMI-A102	computer	Windows XP SP3	172.16.40.0	09:00:09:00:01:12
089f901d-cb58-4055-9652	ACMEincHQ_SW1	switch	Firmware: h.10.38	192.168.0.0	00:16:b9:49:b6:40
e393a902-68fb-4567-b2d1	Modicon M340 BMX P34 20	PLC	Firmware: v2.9	172.16.1.0	00:60:78:00:69:f8

Table 3.1: Example of assets table

3.3 Vulnerability assessment

A *vulnerability* is a bug that could be triggered accidentally or intentionally exploited, and result in a security breach or violation of the system's security policy. This type of bug can be caused by a software/system misconfiguration, or a faulty design or implementation of the infrastructure or a software module. To categorize the various vulnerabilities, this methodology uses the *Common Vulnerability and Exposures* (CVE) standard and the *Common Vulnerability Scoring System* (CVSS) for grading them.

Performing this type of assessment requires an understanding of the priorities of the system. Just as for IT networks, securing the operational network of a thermal power plant means maintaining confidentiality, integrity, and availability (CIA triad), with the only difference being the emphasis given to availability maintenance.

To carry out this part of the study, a *Security Information and Event Management* (SIEM) system must be used. This software compares the asset inventory with a database of known vulnerabilities, and it returns a table of CVEs and relative scores. Also, penetration testing could be used, but is not advised in the context of critical infrastructures as the use of this intrusive technique could compromise the system and make it unavailable.

The output of this phase is a database table resulting from the vulnerability scan, where a vulnerability is a tuple *(ID, asset_id, cve_code, matching_cpes, cve_summary, cve_score)*, such that: *ID* is the primary key, *node_id* is the id of the asset affected by the vulnerability, *cve_code* is the identifier of the CVE, *matching_cpes* is the set of CPEs identifiers that is affected by the vulnerability, *cve_summary* is a description of the vulnerability, *cve_score* is the CVSS score assigned to the CVE.

ID	asset_id	cve_code	matching_cpes	cve_summary	cve_score
4f[...]c1	103b724d-[...]-89d5	CVE-2000-1218	cpe:/o:microsoft:windows_xp:-:sp3:-	The default configuration [...]	7.5
62[...]3b	089f901d-[...]-9652	CVE-2013-6926	cpe:/o:siemens:ruggedcom_system:-:-:-	The integrated HTTPS server [...]	8

Table 3.2: Example of vulnerability table

After this last analysis, also the alerts logs are taken into consideration to better understand the current status of the network. An alert is a tuple *(ID, alert_type, time, description, risk, protocol,*

[2]A NIC in promiscuous mode is a mode that causes the controller to pass all of the content of a network packet to the CPU to "sniff" it.

ip_src, ip_dst), where: *ID* is the primary key, *alert_type* is the code referring to the type of alert, *time* is the date and time (`ISO 8601`) of when the alert was raised, *description* is a human-readable summary of the event, *risk* is a number with precision 1 ranging from 1 to 10, *protocol* is the protocol affected, and *ip_src* and *ip_dst* is the IP of the involved machines.

ID	alert_type	time	description	risk	protocol	ip_src	ip_dst
84[...]b7	SIGN:MALWARE-DETECTED	**2022-03-25** 12:12:27.120	Suspicious transferring [...]	10	smb	192.168.2.0	192.168.1.0
84[...]d4	SIGN:ACCESS-DENIED	**2022-03-27** 14:34:23.97	Unsuccessful login [...]	8.5	smb	192.168.2.0	192.168.1.0
54[...]c5	SIGN:SYN-FLOOD	**2022-03-27** 15:32:21.105	A suspicious [...]	7	tcp	192.168.1.0	192.168.0.0

Table 3.3: Example of alerts table

3.4 Mitigation proposal

To propose a set of mitigations, we need to analyze the vulnerabilities found in the previous step, other than taking into consideration the network architecture the systems employed in it, and also how the network works with the physical process. In OT networks it is common to find legacy systems that enlarge the attack surface. To keep this surface as small as possible, a set of upgrades is required; note that doing a software upgrade, such as from Windows NT to Windows 7, also implies a hardware upgrade. Despite this solution being capable of mitigating a large set of vulnerabilities, it has two main problems:

- *Low scalability.* The project of securing a network comes with a budget and buying licenses and hardware to replace the currently deployed systems is expensive, especially if the number of devices to upgrade ia great.
- *Chances of service disruption.* Usually, IACSs run on proprietary software that can be in its end-of-life or even end-of-support stage. If we upgrade a system that runs a mission-critical piece of software in its end-of-support stage, we do not have any guarantees that this software will still run on the upgraded device.

For the reasons above, the proposals that will be given will not comprehend any system upgrade, but only workarounds, system patching, and network communication re-configurations.

4 Case Study

The presented case study is based on the methodology previously introduced and aims for its verification. The research activity took place between March and April 2022. Lasting a total of 225 hours, the experiment was carried out during an internship in partnership with *ATOS SpA*.

ATOS, is a French multinational information technology (IT) service and consulting company, specializing in hi-tech transactional services, unified communications, cloud, big data, and cybersecurity services. During the internship period, I joined the Big Data & Security (BDS) in evaluating the security posture and monitoring the operational network of eight different thermoelectric power plants belonging to one of the Italian top-players in energy production and management, from now on referred to as **LUX**.

Working in collaboration with skillful partners, the main objective of the project was to implement asset and vulnerability visibility, network monitoring, remediation, and incident response, all without compromising this particularly delicate apparatus.

4.1 Project

Given the need for a better security posture and as a result of the increasing number of cybersecurity threats to various Industrial Control Systems, **LUX** assigned to Atos, as system integrator, the task to perform the required steps to improve the security level of their OT infrastructure.

This task presented various challenges: all of the analyses were performed on a third-party network, so we had no direct access to the network and could perform only agreed operations on it; furthermore, coordination with **LUX** technicians was key to obtain as much knowledge as possible about the deployed systems; also, it is important to note that any communication disruption could have caused fatal consequences, as the scanning, mitigations, and response actions were performed on critical infrastructure and could result in a denial of control on the end service (i.e. energy delivery).

The final goal was to implement a better security posture by offering asset and vulnerability visibility, network redesign, and intrusion detection and response.

4.2 Environment

Since **LUX** operates on a broad area, spacing on all the Italian national territory, there was a need for a system capable of doing a high level of data aggregation, other than offering the possibility to analyze specific security events. Furthermore, as the process of asset discovery was required to be non-disruptive, a set of IDSs with an integrated passive asset-detection capability had to be deployed on the operational networks of the eight thermoelectric plants.

4.2.1 Network sensors

The Nozomi Guardian sensor is an appliance that checks all of the above requirements. In the plants the following systems have been deployed:

- *Nozomi NSG-L Series.* This type of physical appliance has been implemented in all the thermoelectric plants; it has the following specifics:
  - **Maximum Throughput**: 250 Mbps
  - **Maximum Protected Nodes**[1]: 1,000
  - **Storage**: 64 GB

[1] As declared by the vendor : "A node in the Environment represents an actor in the network communication and, depending on the protocols involved, it can be something ranging from a simple personal computer to an RTU or a PLC."

  - **Management Ports**: 1x1000Base-T
  - **Monitoring Ports**: 5x1000BASE-T
  - **Expansion slots**: 4x1000Base-T or 4xSFP

- *Nozomi NSG-M 750 Series.* This sensor has been implemented in the hydroelectric plant;
  - **Maximum Throughput**: 1 Gbps
  - **Maximum Protected Nodes**: 10,000
  - **Storage**: 256 GB
  - **Management Ports**: 1x1000Base-T
  - **Monitoring Ports**: 7x1000BASE-T and 4xSFP
  - **Expansion slots**: 4x1000Base-T or 4xSFP or 4xSFP+

During the experiment, the appliances ran on N2OS v.22.0.0-02111459_10CF8 Nozomi proprietary software.

In order to offer high-level visual of all the operational networks, and to easily monitor all of the power plants, also a Nozomi Central Management Console (CMC) was deployed on a BULLSEQUANA SA10 server, with the following configuration:

- **CPU**: 64xAMD EPYC™ 7662 @ 2.0GHz
- **RAM**: 256 GB
- **Storage**: 2x4 TB HDD, RAID1
- **OS**: Red Hat Enterprise Linux v. 8.0

4.3 Case study: Plant 1

4.3.1 Data gathering

To assess the security posture of **Plant1** the asset inventory and the operational network diagram were requested from **LUX**. Since it is common for asset managers of this type of network to not have a deep understanding of the devices deployed and, as a result, to not have a real asset inventory, the only information we were able to acquire was the network diagram in figure 4.1. While this figure is a reconstruction of the map, the original one can be found in attachment A.

Aiming to enhance the asset visibility, a network scan was performed using a Nozomi *Guardian NSG-L* sensor. This type of appliance offers both active and passive asset inventory, i.e. network scan. Even though an active scan provides more information rather than a passive one, it also injects traffic into the network and, since we were operating on an old and delicate infrastructure, **LUX** imposed us to perform only passive scans.

As we were conscious that **LUX**'s diagram was an extreme simplification of the actual schema, it was decided to proceed modularly by mapping the mission-critical network segments first. As for this thermal power plant, the most critical point in the network was considered to be `VLAN 172.31.0.0/24` and `VLAN 172.32.0.0/24` where the SCADA/DCS module and the attached PLCs used to control the turbines reside. Despite the project's aims to map the entirety of the operational LAN, at the time of writing, the network mirroring is enabled for this segment only.

To perform the scan on such VLAN, the sensor was connected to a switch at the edge of `VLAN 172.31.0.0/24` as in figure 4.1 by setting up a SPAN (Switched Port Analyser) port, granting traffic visibility. Once the appliance was wired to the network, it started sniffing the packets. Performing a deep packet inspection, Guardian uploaded data regarding the assets and the communications on an integrated relational database. It was then possible to query this database via Nozomi's proprietary query language.

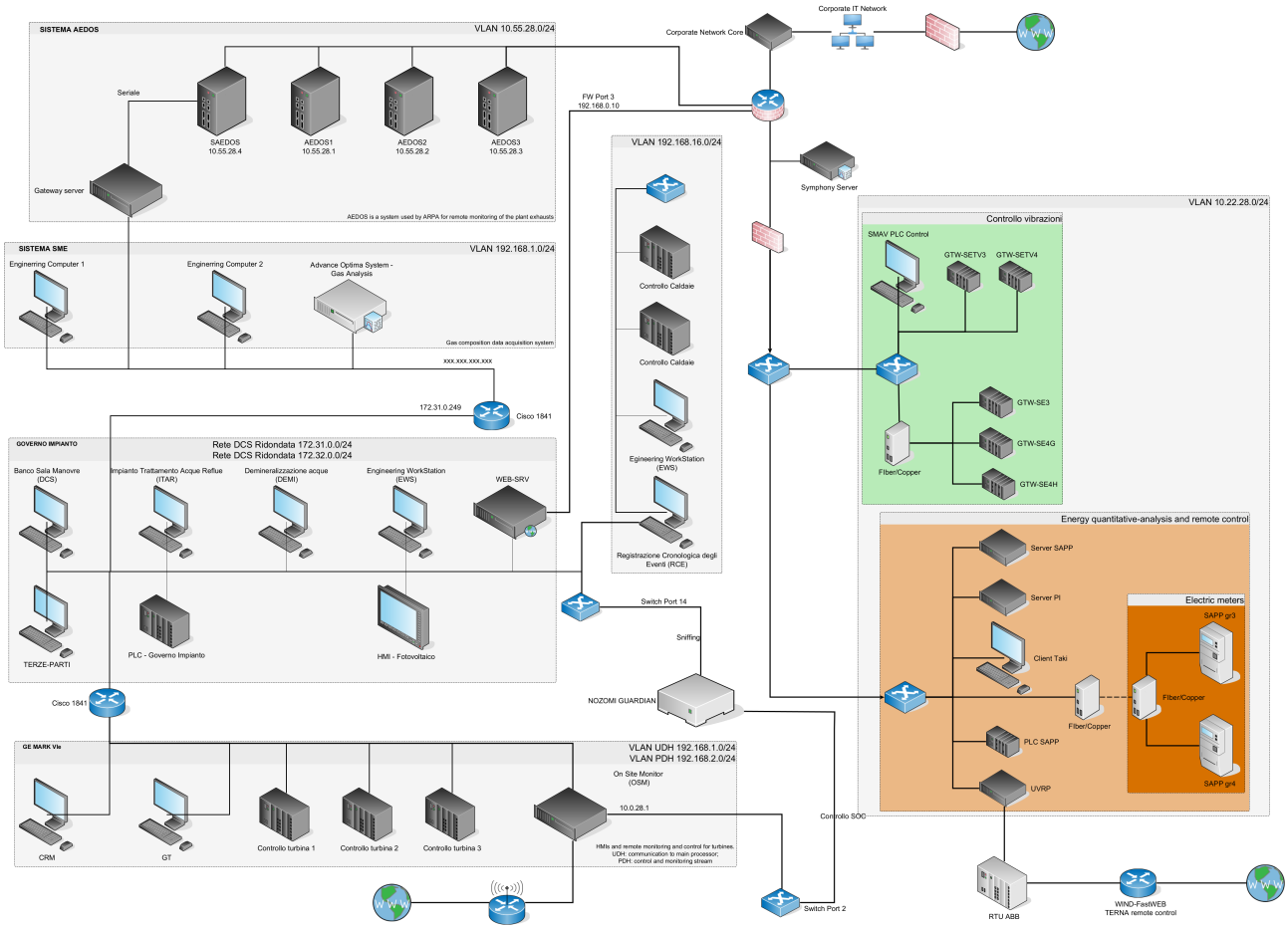

Figure 4.1: Plant1.

4.3.2 Data analysis

Once the sensor was set up, we were able to access data regarding the deployed devices and the communications between them. Using this information, it is possible to *improve asset visibility*, *produce a vulnerability assessment* and *propose a set of mitigations*.

Improving asset visibility

Since we were aiming to assess the security posture of this operational infrastructure, we first needed to draw a more accurate network diagram. To obtain a deeper device visibility, the database was queried to retrieve data regarding the assets visible from `VLAN 172.31.0.0/24` and on `VLAN 172.32.0.0/24`.

The query returned a total of 170 IP addresses belonging to 115 physical devices, resulting in a significant drill-down of the asset visual.

```
-- OSs deployed on 172.31.0.0/24 at first PERA level
node_cpes | join nodes node_id ip
| where joined_node_node_id_ip.level == 1 | group_by cpe

-- OSs deployed on 172.31.0.0/24 at second and third PERA level
node_cpes | join nodes node_id ip
| where joined_node_node_id_ip.level > 1 | group_by cpe

-- Protocols used in the sniffed network traffic
links | group_by protocol
```

Listing 4.1: Examples of query used to retrieve the asset data

During this phase, we expected to find Windows and Linux operating systems for HMIs and servers, and proprietary specific-purpose systems for PLCs and RTUs. As shown in figure 4.2, 58 of the 71 level one IP addresses scanned were assigned to devices operating on Windows NT 4.0 installed, only 7 made use of Siemens proprietary software, while the remaining devices' software was not mapped. . As for the second and third PERA levels, for all the IP addresses, the OSs were mapped as shown in table 4.1.

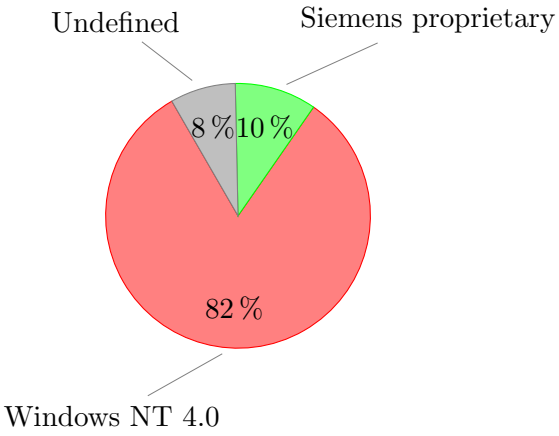

Figure 4.2: Software distribution for the first PERA level.

CPE	*IP Count*
cpe:/o:microsoft:windows_2003:-:-:-	4
cpe:/o:microsoft:windows_server_2003:-:sp2:-	2
cpe:/o:microsoft:windows_server_2003:r2:sp2:-	1
cpe:/o:microsoft:windows_7:-:-:-	1
cpe:/o:microsoft:windows_2000:-:-:-	25
cpe:/o:microsoft:windows_xp:-:sp2:-	24
cpe:/o:microsoft:windows_xp:-:sp3:-	7
Not mapped	24

Table 4.1: Software distribution per IP address for devices residing in PERA second level and above.

Not only the data returned by the query outlined an extensive use of legacy Windows systems, but also that these OSs were deployed on the first PERA level. In addition, to cause a violation of the requirements of the first PERA level, where it's required high system responsiveness, this network infrastructure is affected by a wide range of vulnerabilities.

To better visualize these weaknesses, the database was queried for retrieving data about the used protocols. As shown in table 4.2, the network made substantial use of *NetBIOS* (Network Basic Input/Output System) and *MS SMB* (MicroSoft Server Message Block) protocols to exchange industrial control variables. More specifically, in figure 4.3 the communications using MS SMB are outlined, highlighting the PERA levels 1 to 3 that are represented in green, purple, and blue respectively; grey nodes are level-unclassified. The red and orange links represent communications with alerts raised by the Guardian IDS.

Protocol	*Link Count*
netbios-ns	661
smb	286
browser	207
netbios-ssn	80
dce-rpc	55
lldp	40
opc	33
igmp	16
ssdp	11
telnet	6
modbus	3
others	57
Not mapped	333

Table 4.2: Used protocols.

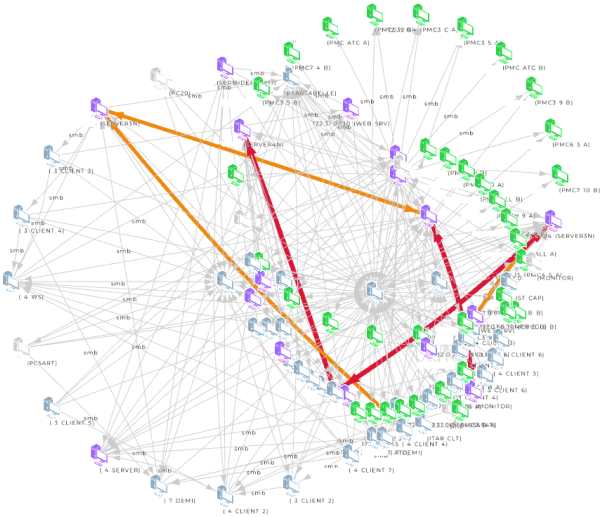

Figure 4.3: Snapshot of links using SMB protocol.

Vulnerability assessment

Once the asset visibility was improved, also considerations on the status of the vulnerabilities could be made. Firstly, the whole network infrastructural design was analyzed. Four main problems were found:

- on the edge of `VLAN 10.0.28.0/24`, a router of the OT network was exposed to the Internet to enable remote monitoring and control through IEC 60870-5-104 protocol. Access was made available through a VPN protected by password-based authentication (SFA). Furthermore, no firewall was deployed to blacklist unnecessary or unwanted protocols, exposing the network to a plethora of threats.
- as in the above point, on the edge of `VLAN 192.168.1.0/24` and `VLAN 192.168.2.0/24`, where the General Electric OSM (On-Site Monitor) was deployed, another router was exposed to enable process variables remote monitoring.
- a violation of *RFC 1918 - Address Allocation for Private Internets* on `VLAN 172.32.0.0/24` where the IP addresses - considered public for the standard - were used as private.
- an improper VLAN segmentation between `VLAN 172.31(/32).0.0/24` and `VLAN 192.168.2.0/24`, as the session logs, highlighted existing links using multiple protocols (dce-rpc, opc, and others) between host belonging to this two network segments. This result in the possibility of lateral movement between the Virtual LANs.

Afterward, having previously outlined an asset inventory and delineated the used protocols, a study of the vulnerabilities affecting the software deployed in the network could be performed. The Nozomi sensor implements an integrated vulnerability scanner using the MITRE standards and open data to asses the asset weaknesses.

The database was queried to list the main CWEs. A large number of CVEs dating prior to the year 2000 were CWE-unclassified, resulting in only 44% of the total CVEs being mapped to a CWE id. These extremely old vulnerabilities were mainly caused by the Windows NT 4.0 and Windows 2000 systems. As shown in table 4.3, the most common class of CWE were *Improper Input Validation*, *Buffer Overflow* , *Broken Access Control*, *Resource Management Errors* , and *Code Injection*.

Common Weakness Enumeration	*Percentage*	*Most recurring CVE*	CVSS2
CWE-20: Improper Input Validation	18.9	CVE-2005-0050	10.0
CWE-119: Buffer Overflow	18.8	CVE-2005-1987	7.5
CWE-264: Broken Access Control	13.3	CVE-2010-0232	7.2
CWE-399: Resource Management Errors	12.2	CVE-2010-0269	10.0
CWE-94: Code Injection	9.9	CVE-2008-4835	10.0
CWE-362: Race Condition	6.1	CVE-2010-0021	7.1
CWE-189: Numeric Errors	5.2	CVE-2009-2511	7.5
CWE-200: Information Exposure	3.8	CVE-2009-0086	10.0
CWE-16: Configuration	1.7	CVE-2008-4609	7.1

Table 4.3: Top CWE.

As it would be impracticable to analyze all of the vulnerabilities affecting the scanned network, only the most recurring ones will be discussed:

- CVE-**2005-0050**. Also known as "Licence Logging Service Vulnerability", this vulnerability affects the system with Windows NT Server 4.0, Server 2000 SP3 and SP4, and Server 2003 software. It is caused by not proper validation of the MS Licence Logging Service (LLS) messages. This service was a feature used to manage Microsoft Server product licenses. If exploited by crafting an LLS message, it can lead to a buffer overflow, allowing an attacker to perform a DoS (Denial of Service) attack or even to remotely execute arbitrary code.

- CVE-**2005-1987**. This CVE affect systems having Windows 2000 SP4, Server 2003, XP SP1, and SP2 installed. It is caused by not proper handling of a received e-mail message header's name. It causes a buffer overflow with the subsequent possibility of remote arbitrary code execution.

- CVE-**2010-0232**. The systems affected by this vulnerability are all Windows OSs from NT 3.1 to Windows 7. When access to 16-bit applications on this x86 platform is enabled, the Microsoft kernel does not properly validate BIOS calls. Crafting a VDM_TIB data structure in the Thread Information Block (Microsoft's Thread Control Block) and then calling the NtVdmControl function to start the Windows Virtual DOS Machine subsystem, can lead to privilege escalation. This exploit cannot be executed remotely or anonymously.

- CVE-**2010-0269**. This vulnerability is also known as the "SMB Client Memory Allocation Vulnerability", and affects the SMB client in different Windows versions released between Windows 2000 and Windows 7, other than Windows Server 2003 and 2008. This CVE is caused by an improper memory allocation for SMB response objects, with the maximum security impact being remote code execution.

- CVE-**2008-4835**. Another CVE was caused by Microsoft SMB. The SMB Server service does not properly handle malformed SMB packets during an NT Trans2 request, resulting in the possibility of remote code execution. The systems affected are different configurations between Microsoft Windows 2000 SP4 and Server 2008.

- CVE-**2010-0021**. The SMB implementation in different Windows releases between Windows 2000 and Windows 7, is affected by poor shared-resource access synchronization. A specifically crafted SMBv1 or SMBv2 Negotiate packet can lead to a DoS attack by system hang, resulting in the operational process Denial of Control.

More specifically, to show the problems derived from the deployment of legacy Microsoft Windows distribution on the first PERA level, the database was queried to retrieve the highest-scoring CVEs affecting the Windows NT 4.0 operational controllers. For the extensive use of SMB - as depicted in figure 4.3 - particular attention was given to vulnerabilities caused by this protocol and leading to *Denial of Service* and *Remote Code Execution*.

The result was two vulnerabilities to remote code execution, that being CVEs 2002-0724 and 2003-0345, and six to DoS attacks, with the most dangerous being CVEs 2003-0345 - as before - and 2005-0045.

As the attack mounting points could be the end systems controlling the plant's actuators and sensors, the successful exploitation of one of the aforementioned weaknesses could result in machinery and process total loss of control. Furthermore, not only the first PERA level presents a large attack surface, but also there are no implemented measures to counter the propagation of an attack based on SMB's vulnerabilities exploit as the entire controllers' variable exchange system is based on this protocol.

After the analysis of the CVE affecting the network assets, also security logs going as back as October 2021 were taken into consideration. While the alerts raised for the OPC and Profinet Network Scan protocols were the result of a PLC function modification on the first, and of a Profinet device misconfiguration on the second, the most alarming logs were the ones related to the SMB protocol.

More specifically, multiple access denied to an SMB share events occurred at a high rate (40 connections every 15 seconds), with source on various nodes in `172.31.0.0/24`, and with the destination on nodes `172.31.0.1` and `172.31.0.2`. Furthermore, those attempts tried to access setup information file `\autoran.inf`, and executable `\Recycled\ctfmon.exe`, other than subsequently tried to establish an anonymous SMBv1 session (operation tied to the EternalBlue exploit). When **LUX** was notified, they responded that the multiple failed accesses were caused by a misconfiguration of the script used to exchange variables between the nodes. The suspicious path in the SMB request and the attempt to establish an anonymous connection remained unexplained.

Mitigation proposals

To harden the network environment the following operations were suggested:

- Firewalls setup on exposed 10.0.28.0/24, 192.168.1.0/24, and 192.168.2.0/24 subnets. The first subnet was exposed to the Internet using a VPN protected by SFA (Single Factor Authentication). The reason for the exposure was to enable Terna to operate through their UVRP (Unità di Valorizzazione della Regolazione Primaria), a server connected to an RTU used to control and monitor the energy delivery. The remote operations were carried out using the IEC 60870-5-104 protocol. Since this was the only protocol used to remotely control the machinery, the implementation of a whitelist containing only the protocols useful for communications with the UVRP was recommended. An analogous solution was proposed for subnets 192.168.1.0/24 and 192.168.2.0/24. These VLANs were used by the General Electric OSM (On-Site Monitor) respectively for turbine monitoring and data processing.

- Revision of network segmentation policies. Since lateral movement is key to successfully attacking an IACS, also redesign of the segmentation between 172.31.0.0/24 and 192.168.1.0/24 was recommended.

- Correction of VLAN 172.32.0.0/24 address.

- Changing protocols for process variable exchange. In order to properly function, an IACS needs the variables describing the current state of the machinery to be shared among the PLCs in real-time and securely. A variables-exchange system using Microsoft SMBv1 does not ensure real-time variable refreshing nor secure communication, as exposed by the numerous vulnerabilities tied to this protocol. A change in the used protocol was advised, in particular, switching to a more conventional protocol like Modbus was suggested. The cost of this solution is very high. This is because changing the protocol for the industrial variable exchange would imply a change in the SCADA/DCS software.

- Since the number of vulnerabilities caused by the indefensible Windows NT 4.0, but also by all the devices using Microsoft's OSs previous to Windows 7, an update of those systems was proposed. The request was rejected as such heavy updates would imply also expensive hardware upgrades and the adoption of a new SCADA/DCS.

4.4 Case study: Plant 2

4.4.1 Data gathering

Following the same methodology used to analyze the first plant, we proceeded to request both an asset inventory and a network diagram to **LUX**. Only the diagram was available. As in the previous case, the delivered network view, pictured in figure 4.4, proved to be generic and not appropriate to proceed with a vulnerability assessment.

The original network graph can be found in attachment B.

A particular related to this graph that may be left unchecked is the presence of physical process' devices, e.g. actuators and sensors (cf. attachment B). Although the presence of these devices in the network map is key to offering a metric for asset criticality, the high-level view of figure 4.4 makes it impossible to produce any evaluation in this sense. Since the industrial network was better maintained than one of the first plants, both **LUX** technicians and Atos consultants approved the implementation of the mirroring on all of the LAN. At the time of writing, of the 9 SPAN ports to be mirrored, only five have been wired to the Guardian sensor. This is because the Nozomi expansion module is still to be connected.

Even if the devices used in the network resulted to be more performant compared to the ones deployed in the previous plant, also this time, only a passive network scan was approved to deepen the network visibility. This proved the distrust felt by the asset managers of this type of system towards active network scans.

Once more, the appliance used was a Nozomi *Guardian NSG-L* sensor.

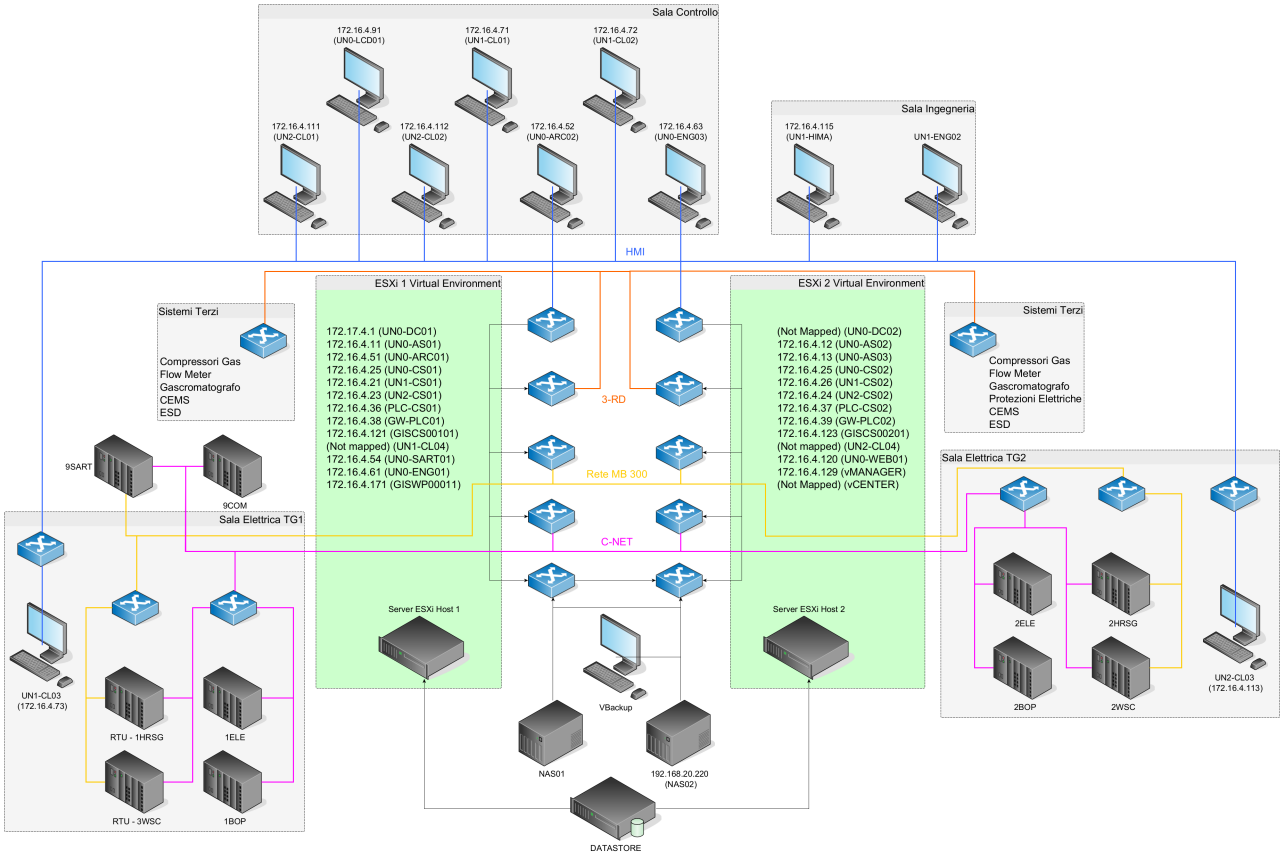

Figure 4.4: Plant2.

4.4.2 Data analysis

With the same goals set in section 4.3.1, we proceeded to *improve asset visibility*, produce a *vulnerability assessment* and *propose a set of mitigations*.

Improving asset visibility

As in the first case study, the network scan produced a drill-down of the communication events and the protocols used within the network. Unfortunately, and different form the previous experiment, the Nozomi appliance was unable to assign a CPE identifier to a significant number of nodes as described in table 4.4.

More particularly, it seems that the identified asset belonged to the HMI, EWS, and the virtualized controller classes; while the other ABB network devices were not fully recognized. The reason why the ABB (ABB Industrial Systems AB) devices were visible only by their IP address or MAC address, was because of the ARP and masterbus300 protocol broadcast messages (that by definition are sent all across the network), while the actual variable-exchange communications between this devices were located in a non-mapped region of the LAN. It is important to note how a protocol-specific active scan, or a complete traffic sniffing, would have solved this problem.

Nonetheless, we proceeded with the analysis of the partially gathered data, while underlining the necessity for a more in-depth view of the network.

As in the previous case study, also the protocols used in the network links were quantified.

Once again, this operational network made extensive use of the MS SMB protocol. Differently from what has been observed in the first case study, in the second plant, the majority of the SMB shares were accessed through SMB versions 3.0, 3.02, and 3.11. This is because the majority of the scanned systems had Windows 8 through Server 2016 installed. Still, some shares were accessed using previous versions of MS SMB, e.g. host `172.20.0.18`, deploying Windows NT 5.1, communicated through SMBv1 with `172.20.0.1` and `10.0.5.8`.

Figure 4.6 is meant to offer a visual representation of the intense use of the aforementioned protocol;

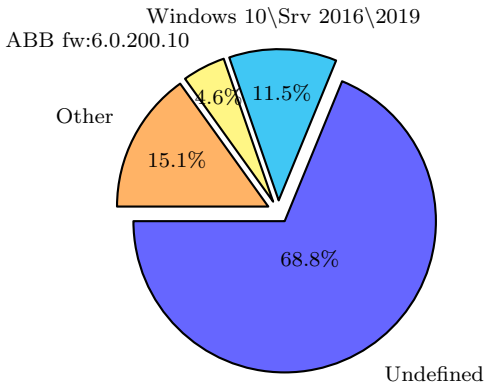

Figure 4.5: OSs distribution for all the assets.

CPE	*IP Count*
cpe:/h:abb:ac_800m:6.0.200.10:-:-	10
cpe:/h:abb:ac_800m_pm864:6.0.200.10:-:-	9
cpe:/o:microsoft:windows_server_2003:r2:sp2:-	4
cpe:/o:microsoft:windows_10:-:-:-	4
cpe:/o:microsoft:windows_server_2016:-:-:-	4
cpe:/o:microsoft:windows_server_2008:r2:sp1:-	3
cpe:/o:microsoft:windows_nt%205.1:-:-:-	3
cpe:/o:microsoft:windows_7:-:sp1:-	2
cpe:/o:microsoft:windows_xp:-:sp2:-	2
cpe:/o:microsoft:windows_8:-:-:-	1
cpe:/o:microsoft:windows_server_2019:-:-:-	1
cpe:/o:microsoft:windows_10:1903:-:-	1
cpe:/o:microsoft:windows_server_2003:-:sp1:-	1

Table 4.4: Detected OS and hardware CPEs distribution for plant 2.

Protocol	*Link Count*
smb	246
llmnr	116
netbios-ns	113
dce-rpc	113
dns	113
rnrp	110
igmp	91
http	86
ssdp	82
cotp	76
Not specified	315
Other	260

Table 4.5: Ten most used protocols.

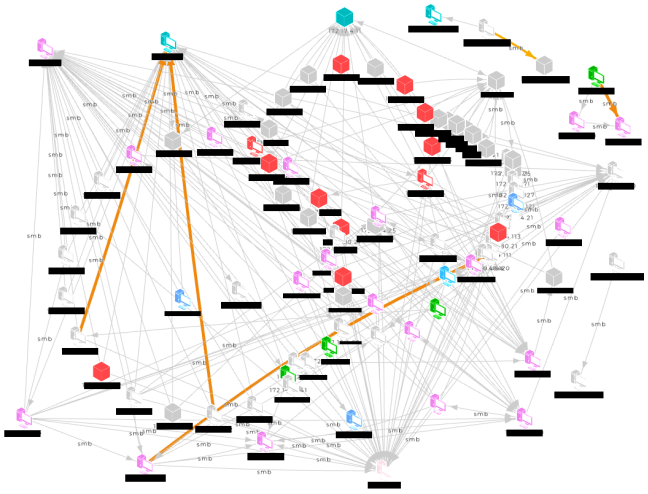

Figure 4.6: Snapshot of links using SMB protocol.

since the asset was not PERA-classified, the colors represent the roles: red for consumers, pink for consumers/producers, light blue for web servers, green for terminals, and grey for non-classified systems.

Since, at the time of writing, the scanned systems deployed in this operational network were still supported, also the state of the installed patches was evaluated. The result of the analysis outlined an irregular patching activity, resulting in systems with a number of missing patches ranging between the dozens and the hundreds. Examples of un-patched systems can be found in table 4.6.

Node	*OS*	*Missing patches*	*Role*
172.16.0.14	Windows XP SP2	254	terminal
172.16.0.12	Windows Srv 2019	16	dns/time server
172.16.0.4	Windows Srv 2016	17	dns server
172.16.0.3	Windows Srv 2016	28	consumer/producer

Table 4.6: Missing patches.

Vulnerability assessment

Even with a limited asset and network view, a set of considerations on the security posture could still be made. Similarly to what was done for the first plant, the network design and practices were

analyzed. The following conclusions were drown:

- despite the presence of an open session between nodes `172.20.4.1` and `192.168.100.27`, breaking `VLANs 192.168.100.0/24` and `172.20.4.0/24` segmentation policy, the risk of lateral movement across the network as a whole were significantly lower than the one affecting the previous power plant.
- as proven by the session logs, node `172.20.154.4` can open a broadcast (`255.255.255.255`) session based on the Microsoft's implementation of the `NetBIOS-ns` protocol. This enables a local attacker to mount NBT-NS poisoning[11]. The same considerations can be made for the LLMNR (Link-Local Multicast Name Resolution) protocol.
- as proven by the monitoring of the network data flow, it is common practice for the asset manager to keep the engineering workstations (EWS) turned on and connected to the network. This could enable an attacker to take control of the EWS and change the connected PLCs' process control logic.

Following the same methodology used in the first case study, an analysis of the asset weaknesses was carried out using the MITRE standards.

The database was queried to list the main CWEs. Despite the asset being more modern than the one of the previous power plant, a large number of CVEs were still CWE-unclassified. This is caused by the presence of multiple vulnerabilities dating back to the year 2000, and the presence of old or unpatched OSs, as shown in table 4.4 and table 4.6. This resulted in only 47% of the total CVEs being mapped to a CWE id.

Differently from what was observed during the assessment performed on the first power plant, the majority of the systems were sharing the same CVEs. This is because, in this case, the distribution of the OSs was slightly more uniform. On top of this, the deployed software was significantly more modern.

Given this more uniform distribution of the CVEs, the highlight of the most dangerous vulnerabilities was preferred instead of the most recurring ones.

Common Weakness Enumeration	*Percentage*	*Most dangerous CVE*	CVSS2
CWE-200: Information Exposure	8.9%	CVE-2021-31976	7.8
CWE-269: Improper Privilege Management	8.4%	CVE-2021-27070	9.3
CWE-119: Buffer Overflow	8.3%	CVE-2019-12655	7.8
CWE-20: Improper Input Validation	6.1%	CVE-2017-0144	9.3
CWE-264: Broken Access Control	4.2%	CVE-2016-3270	10.0
CWE-787: Out-of-bounds Write	3.1%	CVE-2020-1126	9.3
CWE-399: Resource Management Errors	2.6%	CVE-2013-3195	10.0
CWE-94: Code Injection	1.5%	CVE-2012-4786	10.0
CWE-362: Race Conditions	1.2%	CVE-2010-0017	9.3

Table 4.7: Top CWE.

- CVE-**2021-31976**. At the time of writing, the description of this vulnerability is not publicly disclosed. From the public Microsoft's report [6], the CVE is remotely exploitable, has low attack complexity, causes a total loss of confidentiality, and requires no particular privileges. An official fix from Microsoft is available. The vulnerability affects different configurations between Windows 8.1 and Server 2019.
- CVE-**2021-27070**. Also for this vulnerability, the description is not publicly available. The CVE is locally exploitable only, has low attack complexity, causes a high loss of confidentiality, integrity, and availability, and requires basic user privileges and user interaction. An official fix from Microsoft is available [7]. The configurations affected are Windows 10:20h2:*, Windows 10:2004:*, Server 2016:20h2:*, and Server 2016:2004:*.

- CVE-**2019-12655**. This CVE affects a worrying number of Cisco IOS switches' firmware versions ranging from 3.16.8s to 16.11.1. The vulnerability can be triggered by the reading of a specifically crafted FTP traffic. The result is a buffer overflow with subsequent device crash and reload. Obviously, by disabling the switches, the availability of the network and, following, the end service, would be compromised. Official patches and workaround are available. The exploitation of this vulnerability could cause a Denial-of-Service.

- CVE-**2017-0144**. EternalBlue. This is the infamous vulnerability that enabled the Lazarus Group to launch the WannaCry ransomware attack and make it propagate. This CVE affects different Microsoft Windows configurations ranging from Vista SP2 to Windows 10:1607 and Server 2016. More specifically, it is caused by an improper input validation in the implementation of the SMBv1 server, allowing attackers to execute arbitrary code via specifically crafted SMB packets.

- CVE-**2016-3270**. If exploited, this CVE allows local users to perform privilege escalation by crafting an application. It is caused by the implementation of the graphics kernel component and it affects different Microsoft systems between Windows Vista SP2 and Windows 10:1607:*.

- CVE-**2020-1126**. An attacker could remotely corrupt the system memory by convincing a user to open a crafted object through an application that makes use of Windows Media Foundation. The affected OSs range between Windows 10:1607:* and Windows 10:1909:*, and Windows Server 2016:* and 2019:*.

Subsequently the analysis of the vulnerabilities, the alerts logs were taken into consideration. An high number of alerts were raised following a set of technical operations. More precisely, 879 program uploads, 564 variable flow anomalies, 70 firmware change requests, 48 new Modbus and OPC function codes detected and 31 engineering operations alerts were raised.

More alarmingly, web servers `192.168.0.17` and `192.168.0.3` were communicating using HTTP secured by TLS 1.0 protocol, which is considered insecure because of severe vulnerabilities; this raised 26 weak encryption alerts and the exploit of this configuration could result in loss of confidentiality.

A total of 9 TCP-SYN flood alerts where reported involving multiple IP addresses in network segment `172.20.4.0/24`; subsequent investigations concluded that the SYN flood was caused by the restart of nodes `172.20.4.38` and `.39`.

Caused by a credential misconfiguration, 8 alerts were raised by multiple "access-denied" and "unsuccessful-login" events tied to the SMB protocol, involving nodes `172.20.4.131`, `10.10.0.120`, and `10.12.0.38`.

Mitigation proposals

To harden the network and defend the industrial process, the following operations and policy changes were proposed:

- Set up of complete traffic mirroring. As shown by this case study, the incomplete setup of the proposed solution can cause a lack of information in the asset inventory. Be aware that this does not mean that the IP or MAC addresses will not be tracked, but that those addresses will be hard to map to a physical machine. To complete the traffic mirroring, the wiring of the 4 remaining SPAN ports to the sensor was suggested.

- Revision of the EWSs network connection policy. To modify the behavior and feedback of the IACS, technicians operate changes in the PLCs logic through the EWSs (i.e. engineering operations). Thanks to the session logs, it was inferred that the EWSs were kept online even when it wasn't necessary. Since keeping these workstations on the network even when it's not necessary can expand the attack surface, it was recommended to turn off these systems when a change in the industrial logic is not required.

- Implementation of a centralized infrastructure for the delivery of Microsoft Windows patches. In other words, the setup of a *Windows Server Update Services* (WSUS) was suggested. A centralized solution for the update of the systems will enable the technicians to schedule the update operations while keeping only a segment of the power plant shut down for maintenance. Even better, the systems update could be performed during planned maintenance on the IACS industrial systems, to minimize the overall downtime.

- Disabling the retro-compatibility for the older MS SMB versions. As discussed numerous times in this dissertation, this protocol has been proven to be dangerous for the availability of the service. To do so, an upgrade to Windows 7 for all Windows XP systems is needed. The availability of this option should be studied with the asset owner since it is could be achieved with more ease than in the first case study.

- Update the Cisco devices that match the CPEs tied to CVE-2019-12655, or apply an official workaround. With the current level of information, it is hard to determine if the network possesses the required level of redundancy to permit a switch to be set offline for maintenance. A way to fix the vulnerability without turning off the device is by restricting the FTP transaction, as described by Cisco in their CVE-2019-12655 security advise[28].

- Change the alert logs management policy. At the time of writing, the security log is filled with false positives caused by misconfigurations and maintenance operations. Moreover, these alerts are tagged as unresolved. It is important to keep the SIEM log as coherent as possible to facilitate future forensics analysis if any will be needed. To do so, it is advised to tag the already analyzed alerts as resolved and attach a summary to them.

4.5 Case study: Plant 3

4.5.1 Data gathering

Differently from the two previous studies, no asset inventory nor network diagrams were available. For this reason, once again the Guardian sensor was used to improve asset visibility. It must be said that, at the time of writing, the Nozomi's expansion module is still to be connected and, as a result, a limited set of data was gathered during this work.

4.5.2 Data analysis

With the same goals set in section 4.3.1, we proceeded to *improve asset visibility*, produce a *vulnerability assessment* and *propose a set of mitigations*.

Improving asset visibility

Once again, given the choice of **LUX** to not use a via-polling scan, the inferred data over the network traffic were sparse and incomplete. Nonetheless, the passive scan was able to delineate some differences between this infrastructure and the ones of the other case studies.

As we can see from table 4.8, also during this analysis the majority of the matched CPE were the ones belonging to systems running on Microsoft Windows, by far the most recognisable operating system by the sensor. Also, some Cisco and HP switches were recognised.

Furthermore, as in the previous case study, and differently from the first one, the EWs and the HMIs appeared to ran on more recent versions of Microsoft Windows, as described by figure 4.7.

Concerning the used protocols, the most significant of the aforementioned differences is the almost complete absence of the Microsoft SMB protocol, as it has been replaced by the Modbus protocol by design. For reference see table 4.9.

Also during this analysis, the patching status of the devices was studied. Table 4.10 shows some of them with the highest number of missing updates.

As a large set of CVE affecting the system is the result of an old software version being used, the need for a centralized update service became apparent.

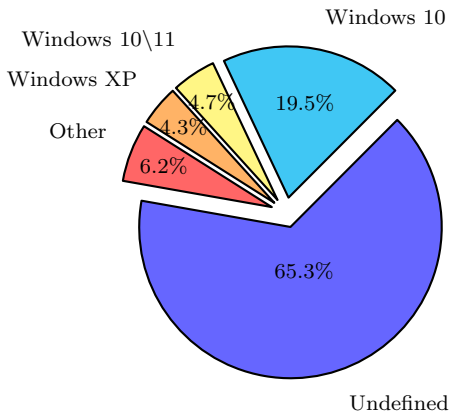

Figure 4.7: Detected OSs distribution for all the assets.

CPE	*Count*
cpe:/o:microsoft:windows_10:-:-:-	56
cpe:/o:microsoft:windows_xp:-:-:-	13
cpe:/o:microsoft:windows_7:-:-:-	7
cpe:/o:microsoft:windows_2000:-:-:-	2
cpe:/o:microsoft:windows_xp:-:sp3:-	1
cpe:/o:microsoft:windows_server_2003:-:sp1:-	1
cpe:/o:microsoft:windows_11:-:-:-	1
cpe:/o:cisco:ios:15.2%284%29e8:-:-	1
cpe:/h:hp:procurve_switch:j9080a:-:-	2
cpe:/h:cisco:telephony:-:-:-	1
cpe:/h:cisco:ws-c2960x-24ps-l:-:-:-	1

Table 4.8: Detected OS and hardware CPEs distribution for plant 3.

Protocol	*Link Count*
http	120
netbios-ns	73
igmp	71
ssh	53
modbus	49
dns	45
ssdp	40
browser	33
rdp	28
llmnr	20
Not specified	74
Other	156

Table 4.9: Ten most used protocols.

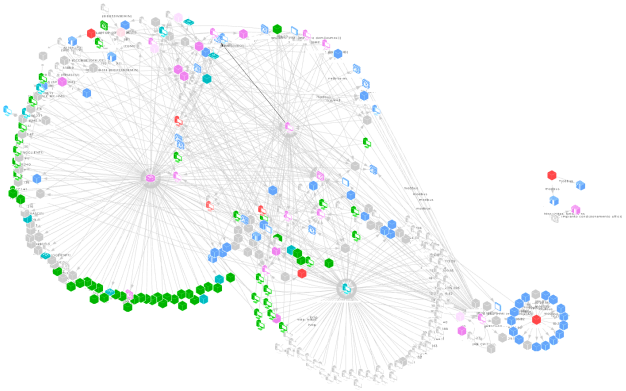

Figure 4.8: Plant 3 OT network structure

Node	*OS*	*Missing patches*	*Role*
10.29.0.60	Windows 2000	125	Not specified
10.29.0.61	Windows 2000	125	Not specified
10.29.0.40	Windows XP SP3	112	consumer/terminal
10.29.0.200	Windows 7	53	producer (HMI)
172.16.0.2	Windows Srv 2016	28	Not specified

Table 4.10: Missing patches.

Vulnerability assessment

Once again, the MITRE standard was used to analyze the weaknesses of the systems used in the network.

As we can see in table 4.11, the Common Weaknesses and Vulnerabilities affecting the asset are nearly identical to the ones that emerged during the second plant assessment. This is due to the similar used technology, and the poor patching policy. For the same reason shown above, also during this study, there was a conspicuous number of CWE-unclassified vulnerabilities, more precisely, 49.4% of them were not classified.

- CVE-**2021-31976**. This vulnerability is the same as found during the second plant assessment.
- CVE-**2020-1412**. This vulnerability affects different versions of Windows 7, 8.1, and 10 and Windows Server 2008, 2016, and 2019. A bug in the memory management of the Microsoft Graphics Components could enable an attacker to execute malicious code remotely. This operation requires user interaction and has a heavy impact on confidentiality, availability, and

Common Weakness Enumeration	*Percentage*	*Most dangerous CVE*	CVSS2
CWE-200: Information Exposure	8.4%	CVE-2021-31976	7.8
CWE-269: Improper Privilege Management	7.6%	CVE-2020-1412	9.3
CWE-119: Buffer Overflow	5.7%	CVE-2019-12655	7.8
CWE-20: Improper Input Validation	5.3%	CVE-2016-7182	10.0
CWE-264: Broken Access Control	3.9%	CVE-2016-3270	10.0
CWE-787: Out-of-bounds Write	3.2%	CVE-2018-8626	10.0
CWE-399: Resource Management Errors	1.4%	CVE-2010-0269	10.0
CWE-416: Use After Free	1.4%	CVE-2014-1776	10.0
CWE-362: Race Conditions	1.2%	CVE-2019-1280	9.3

Table 4.11: Top CWE.

integrity. A vendor solution is available, meaning the systems presenting this vulnerability have not been patched.

- CVE-**2019-12655**. Also, this vulnerability was described in the previous study.
- CVE-**2016-7182**. Another bug of the Graphics component is present in various versions of Microsoft Windows Vista 7, 8.1, 10, Server 2008, and 2012, but also affects proprietary software like Office 2007, 2010, Word Viewer, and Skype for Business. The vulnerability allows an attacker that crafted a TrueType font to execute arbitrary code during the parsing of such a file.
- CVE-**2016-3270**. This bug was also found in the previous assessment.
- CVE-**2018-8626**. This vulnerability affects those Windows systems that are configured as DNS servers and could enable an adversary to remotely execute arbitrary code. An official patch is available.

Among the numerous alerts raised by the SIEM, three incidents were the most interesting.

On March 2022 a host with public IP address sent multiple malformed packets that could be classified as a possible teardrop attack attempt. Even if teardrop attacks are commonly executed on older operating systems, such as Windows 95 and NT, we must keep in mind that legacy software is commonly used in these networks. A check on the destination host was advised.

Until January 2022, multiple attempts at opening anonymous MS SMB shares between two nodes were intercepted. Further investigations revealed that it was caused by a misconfiguration.

Between March 14 and 20, multiple MAC addresses were intercepted by the Guardian appliance. Even if this could be the result of the mirroring of new network traffic, an investigation into these hosts was advised.

Mitigation proposals

In order not only to harden the environment but also to enable the delivery of a more exhaustive assessment, the following operations were suggested:

- Deliver a high-level view network map. Without an exhaustive description of the network structure, and without an understanding of how it interacts with the production process, the proposal of a complete set of mitigation would be not feasible.
- Once again, it is advised to disable the retro-compatibility with MS SMB v.1.
- Check with the ICS vendor for the possibility to implement a protocol whitelist to reduce the number of possible attack vectors.
- Check with the ICS vendor for the possibility to patch the most problematic operating systems.

5 Conclusion

This work offered an overview of the security issues, frameworks, and threats to networks working in the context of industrial process automation, supported by three case studies outlying the major problems. During these studies, a non-intrusive methodology for asset discovery was proposed. To avoid traffic obstruction, Nozomi Networks' appliance, Guardian, was installed in the network to analyze the traffic and automatically infer the property of the asset, offering us in-depth analytics. The choice of this device was also driven by its capability to integrate IDS and SIEM functionalities, other than its ability to conduct passive network monitoring.

Unfortunately, given the low visibility that the asset owner had on these networks, and given the fact that still now the traffic of the networks is not completely mirrored, the requirements of this project have only partly been met. For this reason, the analyzed data are sparse. Nonetheless, the study can still show how the previously described security issues are very much present in real-life systems, the visibility problem above all. At the time of writing, the implementation of this system is being tuned by completing the network mirroring and mapping the devices to the physical process.

At this point in the history of OT networks, given how common it is to find legacy devices deployed, the main viable solution to secure this infrastructure is designing and implementing robust perimeter security and network hardening. To do so, the presented architecture and frameworks are *de facto* standards for engineering network safety systems. Also studying past attacks, knowing the adversary *modus operandi*, and understanding the underlying physical process are key to providing a good security solution.

The Ukrainian war and the recent cyber attacks against its utility infrastructure forced recent and past events to be seen as a national security concern. This can be confirmed by the late establishment of the Italian Army Signal Command Cybernetic Security Unit, and the emphasis given to it during the last months, giving a signal that also in our country things are changing. It is shown in this dissertation how the cybercrime rate related to these critical infrastructures is growing, and how the threats to these systems present themselves as a matter of national security. Finally, we underline the need for a strict *de iure* standardized framework for designing secure ICSs and updating the legacy ones, other than more zealous supervision during its implementation.

After all, *leges bonae ex malis*.

Bibliography

[1] Ujvarosi A. Evolution of SCADA systems. *Bulletin of the Transilvania University of Braşov*, 9, 2016.

[2] Simon Duque Anton, Daniel Fraunholz, Christoph Lipps, Frederic Pohl, Marc Zimmermann, and Hans D Schotten. Two decades of scada exploitation: A brief history. In *2017 IEEE Conference on Application, Information and Network Security (AINS)*, pages 98–104. IEEE, 2017.

[3] Michael J Assante and Robert M Lee. The industrial control system cyber kill chain. *SANS Institute InfoSec Reading Room*, 1, 2015.

[4] Defense Use Case. Analysis of the cyber attack on the ukrainian power grid. *Electricity Information Sharing and Analysis Center (E-ISAC)*, 388:1–29, 2016.

[5] Reynders D. Clarke G. and Wright E. *Practical Modern SCADA Protocols*. Newnes, 1st edition, 2004.

[6] Microsoft Corporation. Server for nfs information disclosure vulnerability. `https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2021-31976`, 2021. Accessed 2022-05-05.

[7] Microsoft Corporation. Windows 10 update assistant elevation of privilege vulnerability. `https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-27070`, 2021. Accessed 2022-05-05.

[8] Daniel DesRuisseaux. Practical overview of implementing iec 62443 security levels in industrial control applications. *USA: Schneider Electric*, 2018.

[9] Samuel East, Jonathan Butts, Mauricio Papa, and Sujeet Shenoi. A taxonomy of attacks on the dnp3 protocol. In *International Conference on Critical Infrastructure Protection*, pages 67–81. Springer, 2009.

[10] IEC SyC Smart Energy. IEC 62443. `https://syc-se.iec.ch/deliveries/cybersecurity-guidelines/security-standards-and-best-practices/iec-62443/`, 2022. Accessed 2022-03-15.

[11] Adaptforward Eric Kuehn, Secure Ideas; Matthew Demaske. Adversary-in-the-middle: Llmnr/nbt-ns poisoning and smb relay. `https://attack.mitre.org/techniques/T1557/001/`, 2021. Accessed 2022-04-28.

[12] Nicolas Falliere, Liam O Murchu, and Eric Chien. W32. stuxnet dossier. *White paper, symantec corp., security response*, 5(6):29, 2011.

[13] Inc. Fortinet. Information Technology (IT) vs. Operational Technology (OT) Cybersecurity. `https://www.fortinet.com/resources/cyberglossary/it-vs-ot-cybersecurity`, 2022. Accessed 2022-03-16.

[14] OPC Foundation. What is OPC? `https://opcfoundation.org/about/what-is-opc/`, 2022. Accessed 2022-03-14.

[15] Igor Nai Fovino, Andrea Carcano, Marcelo Masera, and Alberto Trombetta. Design and implementation of a secure modbus protocol. In *International conference on critical infrastructure protection*, pages 83–96. Springer, 2009.

[16] Sagarika Ghosh and Srinivas Sampalli. A survey of security in scada networks: Current issues and future challenges. *IEEE Access*, 7:135812–135831, 2019.

[17] Kevin E. Hemsley and Dr. Ronald E. Fisher. History of industrial control system cyber incidents. 12 2018.

[18] Erdal Irmak and İsmail Erkek. An overview of cyber-attack vectors on scada systems. In *2018 6th international symposium on digital forensic and security (ISDFS)*, pages 1–5. IEEE, 2018.

[19] ISAGCA. Quick start guide: An overview of isa/iec 62443 standards. `https://gca.isa.org/isagca-quick-start-guide-62443-standards`, June 2020. Accessed 2022-11-03.

[20] Kiuchi M. Li D., Serizawa Y. IEEE/PES Transmission and Distribution Conference and Exhibition. In *Concept design for a Web-based supervisory control and data-acquisition (SCADA) system*, page 32–36, Yokohama, Japan, 6-10 October 2002.

[21] Gaoqi Liang, Junhua Zhao, Fengji Luo, Steven R Weller, and Zhao Yang Dong. A review of false data injection attacks against modern power systems. *IEEE Transactions on Smart Grid*, 8(4):1630–1638, 2016.

[22] OTORIO Ltd. 2022 OT Cybersecurity Survey Report. `https://www.otorio.com/resources/2022-ot-cybersecurity-survey-report/`, 2022. Accessed 2022-03-17.

[23] Michael McFail, Jordan Hanna, and Daniel Rebori-Carretero. Detection engineering in industrial control systems. ukraine 2016 attack: Sandworm team and industroyer case study. Technical report, MITRE CORP MCLEAN VA, 2022.

[24] Palo Alto Networks. The Impact of IT-OT Convergence on ICS Security. `https://www.paloaltonetworks.com/cyberpedia/the-impact-of-it-ot-convergence`, 2022. Accessed 2022-03-15.

[25] Ackerman P. *Industrial Cybersecurity: Efficiently monitor the cybersecurity posture of your ICS environment*. Packt, 2nd edition, 2021.

[26] RSH Piggin. Development of industrial cyber security standards: Iec 62443 for scada and industrial control system security. In *IET conference on control and automation 2013: Uniting problems and solutions*, pages 1–6. IET, 2013.

[27] Marzona A. Pighin M. *Sistemi Informativi Aziendali : ERP e sistemi di data analysis*. Pearson, 3rd edition, 2018.

[28] Cisco System. Cisco ios xe software ftp application layer gateway for nat, nat64, and zbfw denial of service vulnerability. `https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20190925-ftp`, 2019. Accessed 2022-05-06.

[29] 3AG Systems. IOT vs. SCADA. What's the difference? `https://www.3agsystems.com/blog/iot-vs-scada`, 2020. Accessed 2022-03-14.

[30] Bonnie Zhu, Anthony Joseph, and Shankar Sastry. A taxonomy of cyber attacks on scada systems. In *2011 International conference on internet of things and 4th international conference on cyber, physical and social computing*, pages 380–388. IEEE, 2011.

[31] Inc. Zscaler. What Is the Purdue Model for ICS Security? `https://www.zscaler.com/resources/security-terms-glossary/what-is-purdue-model-ics-security`, 2022. Accessed 2022-03-15.

Attachment A Plant 1 network diagram

In this attachment the original schema of plant 1 provided by **LUX** is shown.

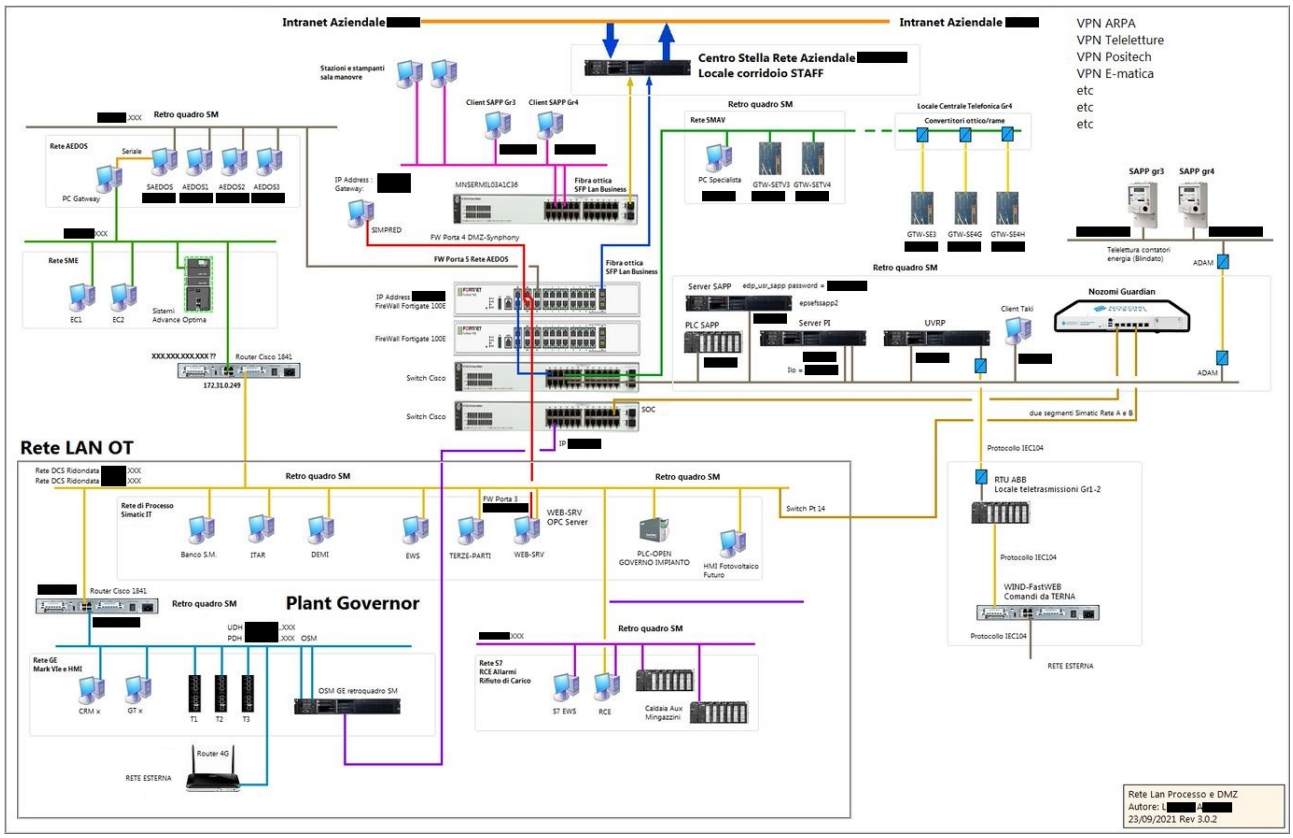

Figure A.1: Plant 1 Original.

Attachment B Plant 2 network diagram

In this attachment the original schema of plant 2 provided by **LUX** is shown.

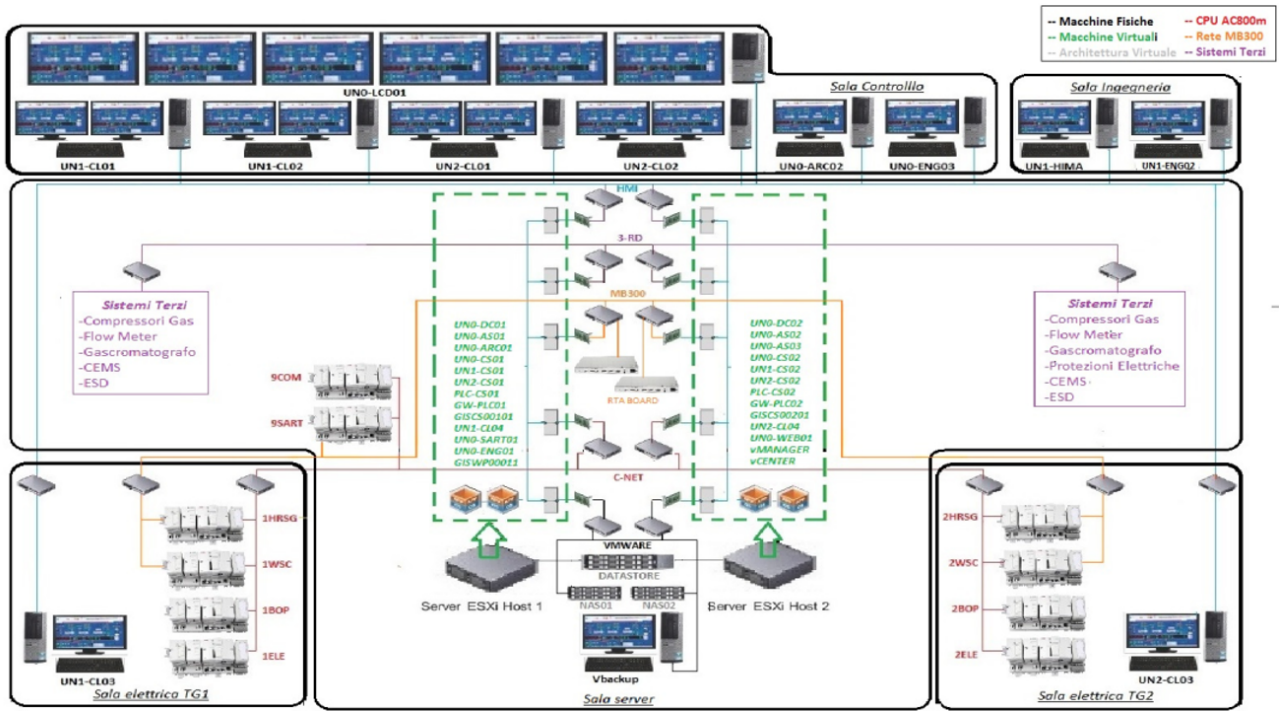

Figure B.1: Plant 2 Original.