

UNIVERSITÀ DEGLI STUDI DI BOLOGNA  
DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA

---

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

---

---

**Industrial Demilitarized Zone and Zero Trust  
cybersecurity models for  
Industrial Control Systems**

---

*Candidato:*  
Silvio Russo

*Relatore:*  
Prof. Michele Colajanni

*Correlatore:*  
Ing. Giorgio Valenziano Santangelo

---

ACADEMIC YEAR 2021/2022





# Index

---

<b>Index</b>	<b>iii</b>
<b>Index of figures</b>	<b>vi</b>
<b>Abstract</b>	<b>viii</b>
<b>Introduction</b>	<b>ix</b>
<b>1 State of the Art</b>	<b>1</b>
1.1 IT/OT Convergence . . . . .	1
1.2 Purdue Model . . . . .	3
1.3 Industrial Demilitarized Zone . . . . .	6
1.3.1 CISCO/Rockwell Automation . . . . .	7
1.4 Zero Trust . . . . .	10
1.4.1 NIST . . . . .	10
1.4.2 Google BeyondCorp . . . . .	21
1.4.3 Zero trust eXtended (ZTX) ecosystem Framework . . . . .	24
1.4.4 VMware NSX Zero Trust Network . . . . .	27
<b>2 ICS cyber-attacks</b>	<b>30</b>
2.1 Security by obscurity . . . . .	30
2.2 MITRE ATT&CK for ICS . . . . .	31
2.3 ICS Cyber Kill Chain . . . . .	33
2.3.1 Stage 1 . . . . .	34
2.3.2 Stage 2 . . . . .	37
2.4 Legacy systems . . . . .	38
2.5 Famous attacks . . . . .	39
2.5.1 Stuxnet . . . . .	39
2.5.2 Industroyer . . . . .	40
2.5.3 Triton . . . . .	40
2.5.4 Pipedream/Incontroller . . . . .	41

---

<b>3</b>	<b>Proposed solutions</b>	<b>48</b>
3.1	Industrial Demilitarized Zone . . . . .	48
3.1.1	Firewalls . . . . .	51
3.1.2	Remote access . . . . .	52
3.1.3	Domain Controller . . . . .	53
3.1.4	Data Historian mirror . . . . .	55
3.1.5	Patch management server . . . . .	56
3.1.6	Digital Twin . . . . .	58
3.2	Zero Trust Architecture . . . . .	59
3.2.1	Access control part . . . . .	61
3.2.2	SIEM . . . . .	68
3.2.3	Segmentation . . . . .	69
3.2.4	Next Generation Firewall . . . . .	70
3.2.5	VLAN Insertion . . . . .	71
3.2.6	Variation for performance improvement . . . . .	72
3.3	Cloud solution . . . . .	74
3.3.1	Network description . . . . .	76
3.3.2	Remote access . . . . .	79
3.3.3	Edge Gateway . . . . .	84
3.3.4	Data Historian . . . . .	86
3.3.5	Security services . . . . .	87
3.3.6	Cloud summary . . . . .	88
<b>4</b>	<b>Implementation</b>	<b>89</b>
4.1	Implementation Access Proxy . . . . .	89
4.1.1	Algorithm . . . . .	91
4.2	Implementation Next Generation Firewall . . . . .	100
4.2.1	Firewall . . . . .	100
<b>5</b>	<b>Validation</b>	<b>108</b>
5.1	Testbed . . . . .	108
5.1.1	PLC . . . . .	108
5.1.2	Compromised IT machine . . . . .	109
5.2	Validation Access proxy . . . . .	112
5.3	Validation NGFW . . . . .	115
	<b>Conclusions</b>	<b>120</b>



# Index of figures

---

1.1	Purdue Enterprise Reference Architecture. . . . .	4
1.2	Zero Trust Pillars. . . . .	13
1.3	Beyond components and access flow. . . . .	22
1.4	Components of the Zero trust eXtended ecosystem. . . . .	25
2.1	MITRE ICS TTP matrix. . . . .	32
2.2	Stage 1 Cyber kill chain. . . . .	34
2.3	Stage 2 Cyber kill chain. . . . .	37
2.4	Pipedream/Incontroller TTPs. . . . .	41
2.5	Pipedream/Incontroller modules. . . . .	47
3.1	I-DMZ on prem. . . . .	50
3.2	Firewalls . . . . .	52
3.3	TIA Portal, accessible devices. . . . .	58
3.4	Zero Trust on-premises architecture. . . . .	61
3.5	Trust Algorithm NIST SP 800-207. . . . .	64
3.6	VLAN insertion Paloalto networks Applying VLAN Insertion in ICS/SCADA. . . . .	72
3.7	PA-5200 Series Performance and Capacities Paloalto Networks NGFW. . . . .	73
3.8	OT alternative. . . . .	73
3.9	Different access technologies. . . . .	76
3.10	Cloud. . . . .	78
3.11	Cloud Authentication and Authorization flow. . . . .	82
3.12	Azure Bastion. . . . .	83
4.1	Trust algorithm Access proxy. . . . .	92
4.2	Login panel. . . . .	93
4.3	Device identification. . . . .	94
4.4	Set Thresholds. . . . .	96
4.5	CVE request. . . . .	96
4.6	Control vulnerable version. . . . .	97

---

4.7	Localization. . . . .	98
4.8	ip-api info. . . . .	98
4.9	Verify device. . . . .	99
4.10	Verify location. . . . .	99
4.11	Netfilter packets flow. . . . .	103
4.12	NetfilterQueue bind code. . . . .	103
4.13	Trust algorithm NGFW. . . . .	104
4.14	packet information. . . . .	105
4.15	address and work hour control. . . . .	106
4.16	authorizer function part 3 . . . . .	106
4.17	drop or accept packet . . . . .	107
5.1	PyModSlave. . . . .	109
5.2	writePLC function . . . . .	110
5.3	Wireshark: Modbus write packet . . . . .	111
5.4	Modbus write packet response . . . . .	111
5.5	Modbus readPLC function . . . . .	112
5.6	Device and location known. . . . .	113
5.7	Device known and location unknown. . . . .	114
5.8	Device and location unknown. . . . .	115
5.9	authorization windows machine . . . . .	117
5.10	TCP connection blocked from the Firewall . . . . .	118
5.11	authorization macOs machine . . . . .	119
5.12	wireshark: authorized connection . . . . .	120
5.13	pyModSlave: PLC register write . . . . .	120



# Abstract

---

Today more than ever, with the recent war in Ukraine and the increasing number of attacks that affect systems of nations and companies every day, the world realizes that cybersecurity can no longer be considered just as a “cost”. It must become a pillar for our infrastructures that involve the security of our nations and the safety of people. Critical infrastructure, like energy, financial services, and healthcare, have become targets of many cyberattacks from several criminal groups, with an increasing number of resources and competencies, putting at risk the security and safety of companies and entire nations. This thesis aims to investigate the state-of-the-art regarding the best practice for securing Industrial control systems (ICS).

We have studied the differences between two security frameworks. The first is Industrial Demilitarized Zone (I-DMZ), a perimeter-based security solution. The second one is the Zero Trust Architecture (ZTA) which removes the concept of perimeter to offer an entirely new approach to cybersecurity based on the slogan ‘Never Trust, always verify’[55]. Starting from this premise, the Zero Trust model embeds strict Authentication, Authorization, and monitoring controls for any access to any resource. We have defined two architectures according to the State-of-the-art and the cybersecurity experts’ guidelines to compare I-DMZ, and Zero Trust approaches to ICS security. The goal is to demonstrate how a Zero Trust approach dramatically reduces the possibility of an attacker penetrating the network or moving laterally to compromise the entire infrastructure.

A third architecture has been defined based on Cloud and fog/edge computing technology. It shows how Cloud solutions can improve the security and reliability of infrastructure and production processes that can benefit from a range of new functionalities, that the Cloud could offer as-a-Service.

We have implemented and tested our Zero Trust solution and its ability to block intrusion or attempted attacks.

# Introduction

---

In the past decade, the concept of perimeter was the basis of a cybersecurity strategy. The defenses of any architecture assumed that after the perimeter was defined, everything inside this perimeter was trusted implicitly. Today this assumption is quite obsolete. The complexity of modern systems and the need for internet connection requires that the “trust” that we give to a specific component must be reviewed. Zero Trust security is a model where different elements of the architecture do not implicitly trust each other, but there is a continuous identification and authorization of all the components. Zero Trust is an emerging but unavoidable paradigm shift for security adopted by the most mature actors, from companies to nations. As we can see, the US government has recently passed a memorandum [15] that obliges all agencies to adopt a security model based on Zero Trust. Zero Trust is a model, a set of principles, and not a specific architecture. It is something that (as any other security solution) must be tailored to the specific needs and goals of the infrastructure to which we want to apply it. More importantly, it is a change of mindset that Security managers and architects must adopt. When dealing with critical infrastructure or ICS in general, the Zero Trust model becomes an essential resource to prevent, detect and react to cyber threats. Those infrastructures are experiencing various security issues and are among the primary victims of cyber attacks.

As reported in a Claroty report[9], 80% of Critical Infrastructure Organizations Experienced Ransomware Attacks last year, and the number of attacks is expected to increase due to new threats and recent political instability. The problem with these critical systems is related to the intrinsic nature of these systems. They were designed to stay up for years or decades, typically with any idea of security in mind, and they rely on older software that is more vulnerable to cyber attacks than others. The security model of these systems was based on the idea of “security by obscurity”. They used obscure protocols and physical isolation from IT network, and because of that, were considered “hack-proof”.

Today, with the convergence of IT and ICS architecture, this security model is no longer valid, making these types of systems among the primary victims of cyber-attacks.

It became clear that for ICS systems, the real question is not “if” an intrusion will take place but “when”. The “castle-and-moat” model is dead, and perimeter-based security solutions can no longer protect infrastructures from threats from inside and outside the network’s perimeter. With the adoption of advanced process control systems such as the Industrial Internet of Things (IIoT) and Industry 4.0, industrial plants require to be connected with the external world. There is a multitude of benefits that organizations can leverage for their stakeholders from greater integration. Sadly, the cybersecurity implications of such integration are less understood.

Our proposal is an innovative Zero Trust solution for the industrial sector, based on a robust continuous Authentication and Authorization system and a per-device segmentation. The architecture, based on an identity-centric security approach, provides complete visibility over the entire network allowing it to protect the infrastructure better and faster identify new threats. It is based on two main components, the Access proxy, and the Next Generation Firewall. The former authenticates, authorizes, and controls any further connection to the company, while the latter is responsible for the internal communications inside the network with complete visibility. All the authorizations are made based on a continuous evaluation of the risk level and the context in which the connection is made, based on a trust algorithm that considers not only the user information but also the device identity and “health.” In this way, we can adapt the security posture of the architecture dynamically by considering several different parameters. The proposed solution is compared with the current state-of-the-art for the security of the industrial sector, the Industrial Demilitarized Zone (I-DMZ). This model allowed for a long time to defend this type of infrastructure well from threats, but now, it is rapidly becoming obsolete and no longer able to protect companies.

The first chapter reported the State-of-the-art of the Industrial Demilitarized Zone and Zero Trust. We will discuss the theme of IT/OT convergence and its effects in the industrial sector from the security point of view, trying to explain the historical reasons that led this sector to such a critical situation, which makes necessary a change in the approach to security.

The second chapter is focused on explaining which are the most used Framework in the industrial sector that help security experts to understand better which are the Tactics, Techniques, and Procedures (TTPs) used by an attacker to conduct intelligence activities. Are also reported brief descriptions of the historical malwares that has affected the industrial sector with a focus on PIPEDREAM, the last discovered one.

The third chapter describes our proposals, the Industrial Demilitarized Zone, Zero Trust, and Cloud solutions so that we can compare them and highlight the benefits that industrial security can derive from an innovative Zero Trust Approach.

The fourth chapter shows how we have implemented the proposed Zero trust solution, the two core components, the Access proxy, and the Next Generation Firewall. In the end, the fifth chapter is about validating the implemented solution.

# State of the Art

This chapter reports the State-of-the-art about the Industrial Demilitarized Zone and the Zero Trust framework. It also describes the current situation of Industrial Control Systems and all the causes that make a radical change in cyber security necessary towards a Zero Trust approach.

## 1.1 IT/OT Convergence

The majority of the problems for industrial security come from a situation that has occurred in the last years, the so-called IT/OT convergence. This refers to the process by which the OT world starts to share technologies and services typical of the IT environment. Gartner defines Information Technology as *“the entire spectrum of technologies for information processing, including software, hardware, communications technologies, and related services”* and Operational Technology (OT) as, *“hardware and software that detects or causes a change, through the direct monitoring and control of physical devices, processes, and events”* [29]. From this definition, it is possible to understand that a characteristic of OT is the ability to interact with the physical world. This is why its security is so critical, involving the safety of the people. As said, OT was historically placed in secure areas, and there was no integration between OT and business networks.

However, this “security by obscurity” strategy can no longer exist, and the line of separation between IT and OT has become increasingly blurred. This convergence process started because the OT world has understood that it can benefit a lot from the technologies that come from IT to improve business processes and the efficiency of the production line. Remote access capabilities can be instrumental in improving the management of an industrial plant. AI/ML techniques have made it possible to extract useful information from the considerable amount of data produced daily from industries, improving productivity and reducing costs. Industrial Internet of Things (IIoT), an extension of IoT in the industrial sector

that allows taking full advantage of the data collection, analytics, and automation features that IoT devices offer, is one of the best examples of this transition and can be considered as the intersection of IT and OT.

All these benefits have meant that this process of convergence proceeds quickly, leaving behind or, sometimes, totally forgetting security problems, threats, and vulnerabilities that have been brought with it. An example of a really dangerous practice, unfortunately not so rare, that this convergence has brought with it is the exposure of PLCs and control systems to the public internet, making them easy to be found by everyone with available public services like Shodan <sup>1</sup>. Everyone can connect to it; they are typically “protected” by very weak passwords that could be rapidly guessed also from a not so skilled attacker, putting at risk the security and safety of the entire infrastructure. This is just an example that shows clearly that the OT world was quite unprepared for the incredibly high number and variety of threats that affect IT.

To make matters worse, the amount of money and the criticality of the businesses around the OT world (that include Critical Infrastructures like communication, chemical factory, healthcare, and energy) have meant that the typical attacker of this type of infrastructure is much more skilled, motivated and with more resources and competencies, compared to the typical attacker in the IT world.

As pointed out by the directive of the U.S. Government [14] Critical infrastructures are a high-priority target for terrorists and state-sponsored Advanced Persistent Threat (APT). They seek to destroy, incapacitate, or exploit critical infrastructure and essential resources to threaten national security, cause mass casualties, weaken our economies, and damage public morale and confidence.

To define an effective security solution, we must identify the goals and core businesses of the infrastructure we want to protect. This is one of the reasons why this convergence has made things quite difficult. The IT and OT world have radically different goals from the security point of view. While IT security focuses on confidentiality, OT usually prioritizes integrity and availability, making it challenging to define priorities and establish an effective security posture.

Without bothering about historical attacks such as Stuxnet or other attacks to the Ukrainian power plants, a representative example of threats that this process of convergence has brought with it, and how much dangerous could be a lack of security on critical infrastructure, is what happened in 2021 to a Florida City’s Water Supply [23]. An attacker manages to intrude remotely into the

---

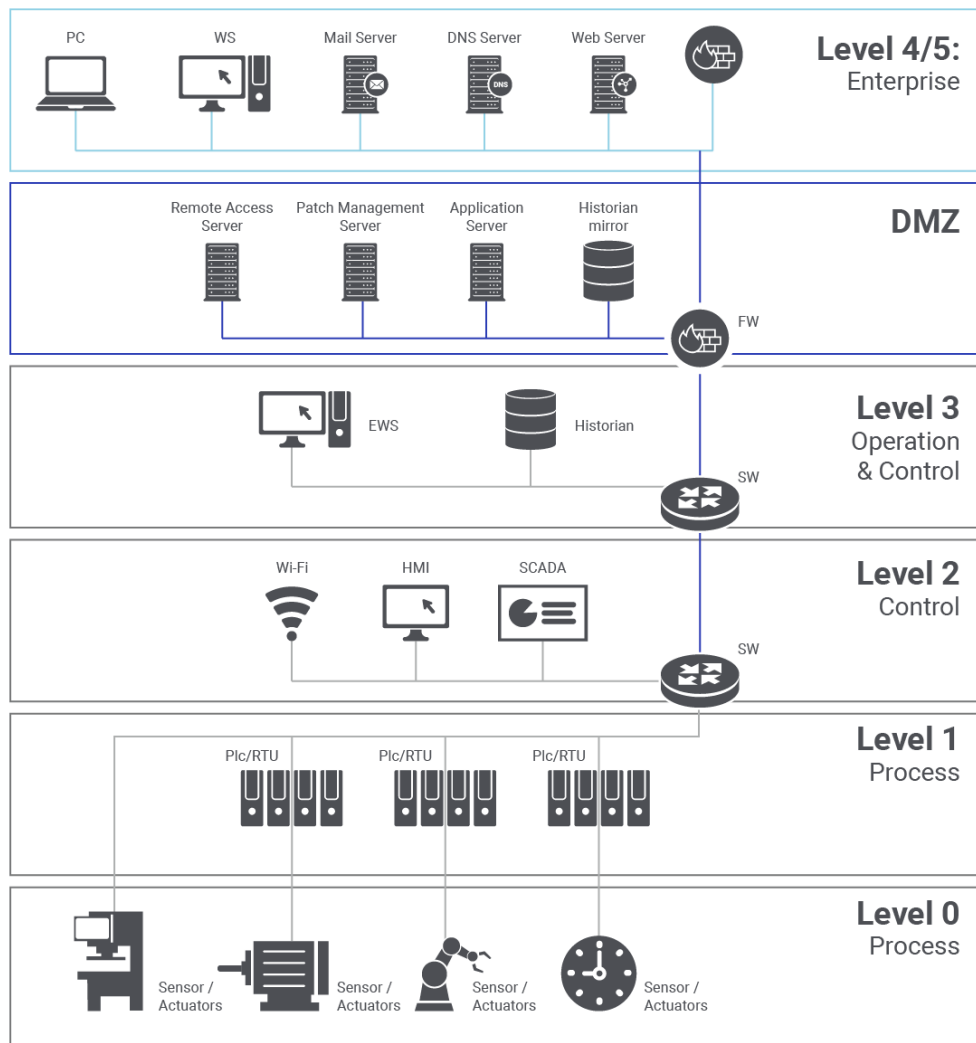
<sup>1</sup>Shodan is the world’s first search engine for Internet-connected devices. url: <https://www.shodan.io/>

infrastructure through TeamViewer, a software for remote control of machines, taking control of an HMI, and thus managing to modify in a few seconds the level of sodium hydroxide (caustic soda) in the water, potentially causing the death of thousands of people. The article states that “the plant used the remote-access software TeamViewer to allow staff to share screens and troubleshoot IT issues, and his boss often connected to his computer to monitor the facility’s systems”. So it is clear that this process of convergence has brought many benefits with it, but it is also crucial that security becomes part of the process to avoid technological progress becoming a weakness for the security of our cities and countries. It is also essential to understand that when we talk about OT and critical infrastructures, the terms “Security” and “Safety” take on a more and more similar meaning, becoming almost synonymous.

New security solutions that go beyond the actual model based on perimeter security are required to protect these new types of infrastructure that “Industry 4.0” brings.

## 1.2 Purdue Model

The Purdue model [74], formally called “Purdue Enterprise Reference Architecture”, is the historical reference architecture for industrial and manufacturing plants.



**Figure 1.1:** Purdue Enterprise Reference Architecture.

The original model didn't have a DMZ level, this is a recent addition, the point of "convergence" between the IT and the OT where are placed all the devices that need to interact with both.

The model separates the industrial architecture into five levels, as shown in the image 1.1.

- **Level 4/5:** this is the level where all the IT devices and services are placed, like the Web server, mail server, and others. Originally this was the only level that was connected with the external world.
- **Level 3:** This level is where the production workflow is managed, and the data are analyzed, it contains the data historian but also databases and all the machines used for the production planning.



- **Level 2:** This is the Control system layer where SCADA, DCS, and all other systems are involved in the production control, like HMI. In the last years, devices of this level are frequently managed remotely, causing an increase in the risk of cyber attacks.
- **Level 1:** Part of this level are PLCs and devices that manipulate physical processes with sensors and actuators. New Industrial IoT devices are placed at this level to increase efficiency. These devices are increasingly communicating directly with their vendor monitoring software in the cloud via cellular networks
- **Level 0:** There are placed the physical devices.

The Purdue model has been considered the reference for industrial architecture for the last thirty years.

Also, from the security point of view, the model has played an important role, helping to define the interactions and the data flows between the different components of the infrastructure, also allowing in the segmentation and segregation process, avoiding flat networks. Also, the implementation of the I-DMZ is based on the division done in the Purdue model by separating the Level 5/4 from the others, filtering all the communications between them.

The idea of the original model was that the OT part of the system would remain "air-gapped", disconnected and inaccessible to both internal networks and the outside world. This is because the firewall is placed only between the IT/DMZ and the OT to underline that all the levels below are implicitly trusted.

Today this assumption is no longer valid [1] with IIoT and Cloud technologies that become more and more widespread, systems and devices can easily communicate not only across layers but directly with the external world. With the use of edge computing devices, vast amounts of data can also be collected at Level 1, processed, and sent directly to the cloud bypassing the hierarchical data flows of the Purdue model. This radical change has led several experts to consider the Purdue model too old and no longer valid for today's infrastructure. This is true if we consider the model as a reference for the organization of the infrastructure. If instead, we consider the Purdue model as an indication to identify the relationship between the different components of the infrastructure and how the data can flow inside the network, it can be handy to help in the segmentation process also for those companies that want to start the transition to new security approaches like Zero Trust.

If we look at our Zero Trust proposal (chapter 3.2), even if there isn't a strict

”hierarchical” separation that follows the Purdue model, this was used to define the interactions between the different parts of the infrastructure and how the data flows between them.

In this way, we can better segment the resources, limiting direct communications between IT and OT and correctly placing PEP and PA.

### 1.3 Industrial Demilitarized Zone

The concept of demilitarized zone (DMZ) comes from the military sector. It is defined as “an area in which treaties or agreements between nations, military powers or contending groups forbid military installations, activities, or personnel”, so it is a territory that acts as a buffer between two zones. As often happens in cybersecurity and computer science, the concept has been taken and adapted.

In network security, a DMZ is defined as “a perimeter network that protects and adds an extra layer of security to an organization’s internal local-area network from untrusted traffic”, also in this case, it is a buffer area between two zones with a different level “Trust” as the Internet and a private network.

Again, in the context of cybersecurity for Industrial infrastructure, the Industrial DMZ (I-DMZ) identifies a perimeter network placed between Informational Technology (IT) and Operational technology (OT) zone that has the role of intermediary for any communication between the two networks.

This division is due to the fact that the OT part of an Industrial network is typically the most vulnerable and, at the same time, the most critical for the business, so with the I-DMZ, we can create a buffer based on the “**Defense-in-depth principle**”, another concept that comes from the military, introducing another layer of security. As stated in the document [24] of U.S. agencies, the Defense-in-Depth Framework is a holistic approach based on detective and protective measures designed to impede the progress of a cyber intruder while enabling an organization to detect and respond to the intrusion to reduce and mitigate the consequences of a breach.

The Defense-in-Depth principle is also defined as one of the fundamental concepts in the IEC/TS 62443 [25], one of the most important standards for industrial cybersecurity.

The I-DMZ is a solution that can be considered part of the Defense-in-Depth Framework, which has as its primary goal the protection through isolation of the OT network.

This need of separation between these two networks is also a consequence of the

different levels of risk between IT and OT and the different priorities. I-DMZ has become part of the Purdue model, as we can see from the image 1.1. This is because, initially, the only part of an industrial infrastructure that could communicate with the external world was the IT part (layer 4/5 of the Purdue model), and in this way, it was possible to guarantee the isolation between IT and OT. We can say that the I-DMZ is the state-of-the-art for security in the industrial sector that still pay a significant "delay" from the security point of view with respect to the IT sector. As previously said, the digitalization of the OT world increases every day, Industrial IIoT, Cloud, Edge computing, and Big data these technologies, despite benefits for the industrial sector that allows creating more efficient and performing processes, require more connections with the external world, exposing OT networks to a new and wide range of threats and vulnerabilities.

In this context, I-DMZ and, in general, perimeter-based security solutions become quite obsolete (even if it's better than nothing), these models require drawing a clear line between Trusted and Untrusted zone, but today, the boundaries of our infrastructures became more and more blurred, they change quickly, making difficult to define clearly and definitively who or what is "trusted".

This is because the work done in this thesis has as its main goal the definition of a Zero Trust solution that, even if it starts to be considered "normal" in the IT sector, for the OT is something new and that just the most mature companies began to implement.

### 1.3.1 CISCO/Rockwell Automation

In the industrial sector, as said, the situation is quite different. The security of OT networks has fallen behind for the reasons previously discussed but also because making changes in an OT network is more difficult and delicate than in an IT environment.

Often stopping the production, even for a few minutes, resulting in a cost that not all companies are willing to pay.

In this sector, an import contribution is given by Cisco and Rockwell Automation, two of the most important companies that produce technologies for the Industries. The two companies regularly publish an important document that includes a collection of tested and validated architectures with best practices and guidelines to build scalable, reliable, and secure industrial infrastructures.

The document, whose last update was published in March 2022 [8], is a reference

for the Industrial Demilitarized Zone (I-DMZ) Framework and offers several solutions to manage the exchange of data between IT and OT network securely, based on the Purdue model.

The document offers several architectural solutions based on the two companies' products, like FactoryTalk by Rockwell, Cisco Adaptive Security Appliances (ASA), or the Enhanced Interior Gateway Routing Protocol (EIGRP), a Cisco proprietary routing protocol. It starts from the initial phases required for a well-designed infrastructure to the implementation to propose a Design Methodology specific to implement I-DMZ-based solutions. The phases described are:

- **Reconnaissance Phase:** used to identify the assets that compose the architecture and how they interact with each other, with particular attention to the interactions between Industrial and Enterprise Zone that, following the I-DMZ model, must pass through the DMZ.
- **Architectural Phase:** after the reconnaissance phase is essential to propose a high-level solution to validate the assumptions that have been made before starting to work on the real system.  
even if it is not underlined in the document, this is a critical phase to understand if the resources are sufficient for the proposed solution.
- **Technical Design Phase:** there is the moment where the technical solutions, which will then be implemented, are chosen to meet the requirements defined in the previous phases.
- **Implementation Phase:** is the moment when the proposed solutions are brought together, tested, verified, and validated.
- **Maintain Phase:** this is the Operational phase where the infrastructure is monitored and eventually fixed. This is one of the most critical phases and, at the same time, one of the most neglected phases.

The document is very vast and detailed and offers useful solutions to implement and manage an important part of an Industrial Infrastructure as the Remote Access Services, Segmentation, Data Historian, and Firewalls placement and management.

In particular, the document defines best practices for implementing two important components of an industrial network, the Domain Controller and the Data historian, which are considered the two historical issues for an I-DMZ implementation.

- **Domain Controller:** The solution proposed by CISCO/Rockwell is based on the implementation of the DC in a single domain with multiple sites, in this way, it maintained a single identity and access policy repository for all employees in a company. It implemented a bi-directional replication between the Enterprise DC and the Industrial Zone DC. An AD administrator should be able to create, delete and update accounts in the Industrial Zone, and the changes will be replicated in the Enterprise Zone and vice versa. Another proposed solution to manage Domains is to use the “**Organization Domain Forest Model**”, also described by Microsoft [48]. In this solution, we define a Forest Root Domain, and several groups, each of these controls its sub-domain allowing to manage some aspects like:

- Management of domain controller operations
- Configuration of domain-wide settings
- Configuration of data-level administration
- Management of external trusts

There is also the Domain owner that has authority over the entire domain as well as access to all other domains in the forest, and for this reason, this role must be assigned to a really trusted user.

- **Data Historian:** For the placement of the Historian, the recommendation is to have two Historian servers installed, one in the Corporate network and one in the Control network, the interaction between them is managed by a so-called PI-to-PI Interface placed in the I-DMZ. This interface pulls predefined data from the Historian in the Control Zone and pushes the data to the Historian in the Industrial Zone.

Another important part of the document talks about policies required to have a well-defined security posture, regardless of the security solution adopted:

- **Defining Roles:** everyone in the company should be assigned a role to define and manage the credentials efficiently, allowing to implement a Role-based access control (RBAC) and the Least Privilege principle, with a well-defined separation of roles within the company.

- **Data Classification:** Identifying and classifying the data produced and exchanged inside a company is crucial to determine a security strategy. We need to know the “value” that these data have for the company to understand the level of risk associated with these data and which are the possible threats that will need to be addressed.

The solution adopted by Cisco and Rockwell is mainly oriented toward selling their products. However, it offers many ideas for the implementation of an Industrial network, with particular attention to the security part, based on fundamental concepts underlying each security solution.

However, the document is focused on implementing security solutions that have as the main goal the protection of the communications between the IT and OT part as if the threats can come only from the IT. As we have said, this is a wrong assumption, and threats can also come from other sources. This is because we need a different approach to secure industrial infrastructures.

## 1.4 Zero Trust

This New approach that we need to protect our industrial infrastructure through the Industry 4.0 revolution is called “Zero Trust”.

### 1.4.1 NIST

#### **who, what, when, where, and how**

As we can read from the title, this thesis is focused on the Zero Trust Architecture (ZTA) framework and how it can be applied to the Industrial sector. The Zero Trust security model is a set of principles and strategies based on the awareness that cyber threats exist and can originate from outside but also inside the “perimeter” of traditional networks.

From this assumption, the Zero Trust model embeds strict and coarse-grained monitoring to any access to any resource. Data becomes the core of the security model to drive the choices about security. Policies and procedures are crucial to define who, what, when, and how these data could be accessed. The least privilege access is the underlying principle leading any interaction; every access decision must be evaluated all the times, minimizing as much as possible any assumption of “Trust”.

Who can access what resource is a crucial question for any Authorization and

Authentication protocol but in Zero trust, the assumption behind all decisions is that all requests for resources are malicious. Because of that, also, from where and from which device (who) someone is trying to access a certain resource must be considered to allow or not access to a certain resource. Also, when access is granted, it is necessary continuous monitoring of the behavior of the user. This obsessive monitoring and control of everything that happens in the systems require technologies but, more important, policies and procedures that can dynamically, in an automated way, evaluate the state of the systems. So it is clear that Zero Trust is not a “plug and play” technology. It is a mentality, a goal to be pursued, something that requires a significant effort in terms of resources and competencies and a certain level of maturity to be implemented.

### Tenets

Give a general definition of Zero Trust architecture is not so easy. It is a complex model which can be adapted to several, even very different from each other, environments and infrastructure, and it is, therefore, necessary, like everything in cybersecurity, to tailor the model to the specific needs of the implementation. Nevertheless, the NIST [55] define Zero Trust Architecture with five fundamental tenets.

- **Never Trust, Always Verify:** Never Trust, Always Verify is the idea behind any design decision in the Zero Trust security model. As said before, any trust assumption should be avoided, but it is clear that “Zero” Trust is an ideal situation that can only be implemented by companies and entities with a huge number of resources and an already established security infrastructure, is this the case of big companies or government agencies. All the other smaller company that wants to deploy a Zero Trust Security model should evaluate which are the priorities of their business and start a path to develop competencies and technologies to implement this model.
- **Assume a Hostile Environment:** The assumption that malicious personas are both inside and outside the network forces to monitor any user, device, network, or environment that is considered as untrusted even if it has passed all the authorization and authentication mechanism. All the operations are strictly monitored to ensure that the resources are used correctly for that specific user, looking for suspicious action that must be quickly detected and analyzed.

- **Assume breach:** Especially in Critical infrastructure, the number of cyber attacks is sadly high, and because of that, the recovery time and fault tolerance are fundamental to assure the continuity of the service. Assume that the system is constantly under attack allows reducing the detection and response time, providing action to isolate and stop the attack.
- **Scrutinize Explicitly:** The right to access a resource is given after establishing a certain level of confidence according to several factors and based on defined policies. Policies can change dynamically to adapt to the current situation of the environment. The number of factors to take into account to decide whether to grant access or not to a resource is crucial for the level of security that we can grant. At the same time, the more the number of factors, the more resources that the system needs to manage access to all the resources without causing bottlenecks in the system. Because resources are limited in most of the implementation, also, in this case, it is crucial to determine which are the priorities.
- **Apply unified analytics:** In a Zero Trust Architecture became fundamental to centralize the collection of data and apply analytics for all the assets of the infrastructure, including information about the behavior of the users to understand which are “common” behaviors thus managing to identify suspicious situations.

## Pillars

A pillar is a key focus area to implement Zero Trust, subdivide architectures in a layered structure, simplify the design of the infrastructure, and make it more flexible to implement zero trust controls.

It allows us to better identify the technologies and capabilities necessary to implement the architecture.



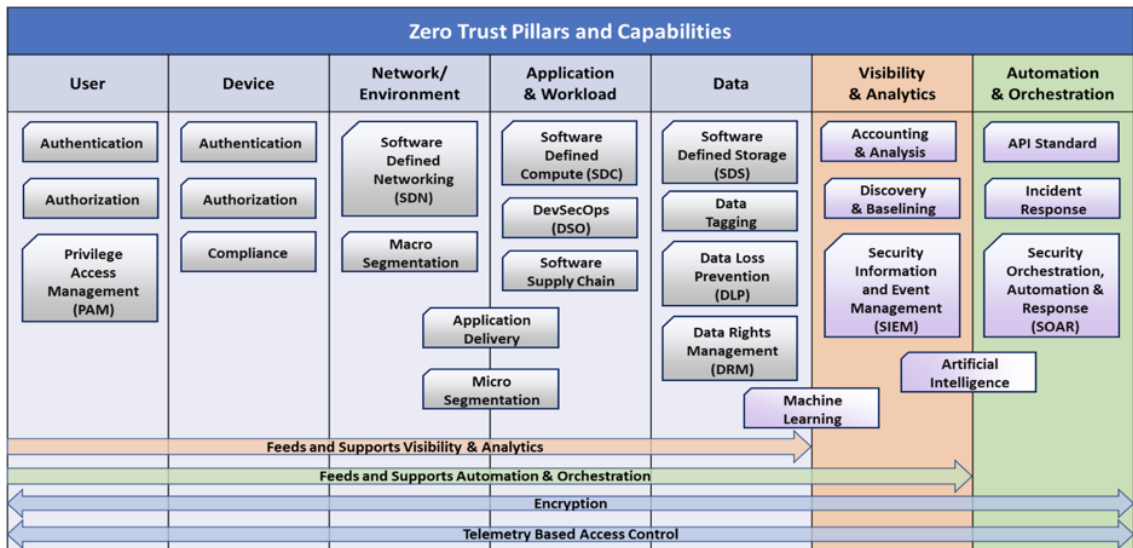


Figure 1.2: Zero Trust Pillars.

- User:** Is crucial to secure and limit any person, non-person, and federated access to DAAS (Data/Asset/Applications/Services) using Multi-factor Authentication (MFA) and continuous Multi-factor Authentication (CMFA). Organizations need the ability to enforce those mechanisms to govern users' access and privileges.
- Device:** In a Zero Trust approach, the ability to identify, authenticate, authorize, inventory, isolate, secure, remediate, and control all devices became crucial. As for users, policies based on a whitelist approach are widely used to control access to DAAS. Assessment should be conducted for any access request to monitor if the device is patched or if there are symptoms of compromise.
- Network and Environment:** Segmentation is an important tool to limit intrusion in the systems. There are different implementations of Zero Trust that follow different principles like Micro-segmentation, Software-defined perimeter (SDN). In our case, micro-segmentation allows us to protect well and control DAAS, prevent lateral movement, control privileged access and manage internal and external data flows.  
 Managing and controlling the network/environment becomes even more important in those cases of “off-premises” situations that require a more fine-grained access restriction.
- Application and Workloads:** Talking about application and workloads, we include systems and services on-premises but also Cloud applications and

services. Zero Trust became even more important when dealing with hybrid or totally on Cloud solutions. With these solutions, data are spread over several data centers, and requests flow in and out from the organization's internal network, thus increasing the risk of attack. Also developing secure code is important and can be exploited by the DevSecOps development practices to secure applications from inception.

- **Data:** Data protection is the core and one of the main goals of any security model, in the case of Zero trust, the model is data-centric. All the design choices are based on securing data, applications, assets, and services, so it became critical to understand which data the organization manages and categorize it to provide a successful and efficient implementation. Solutions such as DRM (Digital Right Management), DLP (Data Loss Prevention), Software Defined Storage, and granular data tagging are relevant in protecting critical data.
- **Visibility and Analytics:** As said, the Zero Trust model is based on dynamic policies to determine the priorities. It is crucial to have a deep understanding of the system's performances and behaviors to change policies and make access decisions in real time. For those purposes, information coming from sensors, telemetry, and other monitoring systems are pooled and analyzed in real-time from a centralized entity to extract data and understand the behaviors of the system. Analytics can go deeper, inspecting the packets themselves to accurately discover traffic on the network and observe threats that are present and orient defenses more intelligently, also trying to avoid bottlenecks.
- **Automation and Orchestration:** Because of the complexity of modern architecture with a huge amount of data to manage and the necessity to adapt the system to sudden changes, the use of automation and autonomous orchestration, typically associated with AI techniques, becomes an irreplaceable tool to implement a Zero Trust architecture, security orchestration integrates Security Information and Event Management (SIEM), and other automated security tools play a key role. Implementing and managing these systems require a certain level of resources and competencies. This is because understanding the organization's maturity level is important to clarify which is the best way to implement a Zero Trust model or to identify a path that can lead to implementing a Zero Trust model.

## Logical components

There are several components that compose a Zero Trust Architecture.

Clearly, these components are just “logical,” and any implementation can be done differently by dividing one single logical component into several physical components or treating two different logical components as one, following the needs of the specific implementation. In accordance with the design’s choices, those components can be operated on-premises or in a Cloud-based solution adapting all the architecture accordingly.

- **Policy engine (PE):** this is the component that is responsible for the decision of granting or not access to a specific resource for a given user. It is based on the enterprise policies but, even more important, it is influenced by what we can call “context”. A context is defined from all the information coming from different sources, like the Threat intelligence system, that gives information about the current situation to define better the level of risk of the specific operation. Those information are inputs for the Trust algorithm to grant, deny or revoke in real-time access to the resource.
- **Policy administrator (PA):** this is the component that “physically”, based on the decisions of the Policy engine, establishes or shuts down the communication between the user and the different resources. Is responsible for the generation of any credentials or access token used to authorize or authenticate the user. If the session is authorized and the request authenticated, the PA configures the PEP to allow the session to start. If the session is denied, the PA signals to the PEP to shut down the connection.
- **Policy enforcement point (PEP):** this is the device that receives directives from the PA and executes the commands in real-time. Also, this component can be composed of several agents placed in different machines or as a single gate that manages and controls any resource’s authorization and authentication. If we want to use the concept of perimeter, the PEP is the device that divides the trusted zone from the untrusted one.

The document also describes other components that cannot be considered core components as the previous one but are important to help to define the context and create a real Zero Trust implementation.

- **Continuous diagnostics and mitigation (CDM) system:** this is the component that gathers the information about the asset’s current state and

allows to maintain patched and updated resources. It is the component from which the Policy engine retrieves the information to decide if allow or deny the connection to a resource accordingly to the state of the device that is making the access to the resource, controlling if it is updated or not.

- **Industry compliance system:** this is the system used to ensure that all the compliance requirements are satisfied.
- **Threat intelligence:** as the CDM, this component helps the policy engine to make a better decision, takes information from several sources, and provides insights about newly discovered attacks, vulnerabilities, or new malware. Also, in this case, all those information are aggregated to define a better context.
- **Network and system activity logs:** another crucial component that can be single or combined with other components responsible for the aggregation of all the logs of the system.

A centralized log management is crucial to make it easier not only to have some feedback on the security posture of the architecture but also, in case of a cyberattack, for the forensics phase, to understand better what happened and speed up the Response and Recovery phases.

- **Data access policies:** this component is responsible for storing the attributes, rules, and policies about access to enterprise resources. This set of rules could be encoded or dynamically generated by the policy engine. These policies should be based on the organization's defined mission roles, needs, and priorities.
- **Enterprise public key infrastructure (PKI):** as better described in the following section, this is the component that manages all the certificates issued by the enterprise
- **ID management system:** It is responsible for creating, storing, and managing accounts and identity inside the company
- **Security information and event management (SIEM) system:** similar to the Network and system activity logs, this component collects security-centric information for later analysis. This data is then used to refine policies and warn of possible attacks against enterprise assets.

## Trust Algorithm

Also for the Trust algorithm, the NIST has identified four different categories:

- **Criteria vs Score-based:** A criteria-based TA(Trust Algorithm) assumes a set of qualified attributes that must be met before access is granted to a resource or an action (e.g., read/write) is allowed. The enterprise configures these criteria and should be independently configured for every resource. Access is granted, or action applied to a resource only if all the criteria are met. A score-based TA computes a confidence level based on values for every data source and enterprise-configured weights. If the score is greater than the configured threshold value for the resource, access is granted, or the action is performed. Otherwise, the request is denied, or access privileges are reduced (e.g., read access is granted but not write access for a file).
- **Singular vs contextual:** A singular TA treats each request individually and does not take the subject history into consideration when making its evaluation. This can allow faster evaluations, but there is a risk that an attack can go undetected if it stays within a subject's allowed role. A contextual TA considers the subject or network agent's recent history when evaluating access requests. This means the PE must maintain some state information on all subjects and applications. However, it is also much more effective in detecting an attacker that uses subverted credentials to access information with a pattern that is atypical of what the PE sees for the given subject. This also means that the PE must be informed of user behavior by other PEPs, like the NGFW. Subject behavior analysis can provide a model of acceptable use, and deviations from this behavior could trigger additional authentication checks or resource request denials.

## Variations of ZTA

There are several ways to implement a Zero Trust Architecture, but the NIST [55] has defined three main variations.

- **ZTA Using Enhanced Identity Governance:** in which identities are the key points to create policies based on the access privilege given to a subject, this model is used for enterprises with several cloud-based services like SaaS.
- **ZTA Using Micro-segmentation:** [75] Based on placing a set of resources in a network segment and protecting them with a PEP that monitors all the

interaction. So the core of this solution is the PEP that needs to be capable of adapting to the new workflows or eventually to a threat.

- **ZTA Using Network Infrastructure and Software Defined Perimeters:** in this case, we use a software solution defining an overlay network (level 7) to implement the policies for the ZTA.

### Threats associated with zero trust

Security is all about risk, risk identification, risk tolerance, evaluation, and treatment. What is clear is that it does not exist a security solution that can reduce to zero cybersecurity risk, and the zero trust model is no different. Nevertheless, with existing cybersecurity policies and guidance, identity and access management, continuous monitoring, and general cyber hygiene, a properly implemented and maintained ZTA can reduce overall risk and protect against common threats.

Otherwise, the ZTA is also affected by specific threats; in particular, a ZTA is composed of different core components like PE, PA, or PEP that could become a single point of failure for the entire architecture. The Nist<sup>[55]</sup> document identifies the different types of threats that can affect a ZTA:

- **Subversion of ZTA decision process:** as said before, the decision process that authorizes or denies access to a resource is managed by the PA and the PE, this means that these two components should be carefully configured and managed.  
If, for example, an administrator changes the behavior of these two components, the entire infrastructure can be considered compromised.

- **Denial-of-Service (DoS) or Network Disruption:** these types of threats can affect any infrastructure, the difference in a ZTA is that a DoS attack or a route hijack disrupt or deny the access to a PEP, PE, or PA it can affect all the enterprise operations blocking all the communication to the internal resources. A possibility to mitigate this threat is given by Cloud hosted solution. All major Cloud actors offer several possibilities to easily replicate these components over different machines to make the architecture more reliable and resilient to these types of attacks.

This can also mitigate the risk that the internal resources are not reachable from the PA that cannot configure the path to the specific resource.

- **Stolen Credentials/Insider Threat:** another widely used technique to intrude in a network is to steal the credentials of a user.

With the multi-factor authentication (MFA) method and Zero Trust approach, we can limit the risk of this threat even if it remains.

Because of that is important to define the roles and the authorizations for each resource. In this way, even if an attacker manages to steal credentials, the number of compromised resources is limited.

As shown in this thesis, a context-based Trust algorithm is one of the most powerful defenses to this type of attack.

- **Visibility of the network:** in any security solution, the visibility of the network traffic is crucial to understand the state of the architecture and to have the ability to detect and respond fast to an eventual attack.

There are some cases in which the traffic is encrypted, and we cannot inspect it, dramatically increasing the risk of an undetected attack. Several Machine Learning (ML) based techniques can be used to analyze the encrypted traffic to avoid this threat.

- **Reliance on Proprietary Data Formats or Solutions:** another significant threat is related to another well-known problem, that is the lock-in, implement a ZT solution requires a lot of effort, competencies and advanced technology, and, commonly, companies outsourcing some service relying on a third-party provider.

If the provider has the same issue, the migration to another one is complex or, sometimes, impossible; this can be considered as a vulnerability.

So it is crucial that the choice of the service provider is made carefully and keeping in mind the cost and other factors like security and reliability.

, In any case, trying to be as vendor independent as possible is a good choice.

- **Use of Non-person Entities (NPE) in ZTA Administration:** more and more often employed Artificial intelligence and software-based agents to manage the network's security.

In a ZT solution, these components need to interact with the PE and PA; this means that they need to authenticate themselves. It is a problem and an open issue that can put the architecture at risk.

## Migrating to a ZTA

In recent years, with the significant increase in cyber-attacks and the consecutive awareness of companies about cybersecurity, ZTA became a “Buzzword”, and like any other buzzword, there are many vendors that use ZTA to sell any security

technology. To avoid this problem is essential to clarify that ZTA is not a single technology or something that can buy and plug-in into a network. ZTA is a journey, a necessary effort that requires competencies, resources, policies, and time to be implemented and consolidated within the company. It is not something that can be implemented overnight, and, even more important, does not exist a single ZTA. Like any other cyber-security solution, it is crucial to tailor the architecture to a company's specific needs and maturity level.

Talking in general, we have two main solutions for implementing a ZTA.

- **Pure ZTA:** There are few cases, like a start-up, that can implement a Zero Trust Architecture from scratch, in these cases, the implementation is clearly “easier” and faster, is necessary to identify the operational workflow and define which are needed for the implementation phase.
- **Hybrid ZTA:** As said are rare companies that can implement a ZTA with a greenfield approach, in most cases, a company has an already established architecture and decide to start to migrate to a ZTA solution. Typically, the approach is to move one business process at a time to evaluate the impact and restore normal behavior in case of problems. Indeed the migration can create problems with the workflow making it difficult for employees to access services.

As we can read in the NIST 800-207 publication [55], there are several phases for correctly migrating to a ZTA:

1. Identify Actors on the Enterprise: to implement the PE and all the procedures for the authentication and authorization, we need to know human and non-human actors that operate in the company.
2. Identify Assets Owned by the Enterprise: as said, identifying the device from which a user interacts with the company's resource is critical to limit the risk, especially in the case of remote connections.
3. Identify Key Processes and Evaluate Risks Associated with Executing Process: as in any other security framework, to define policies and a security strategy, we need to know the key process and the business core of the company that we want to protect, only then we can define the risk level associated to a specific resource and who can access it.
4. Formulating Policies for the ZTA Candidate: after that, the actors, assets, and processes have been identified, we need to define the policies and



procedures to manage those resources. We also need to determine a candidate workflow to implement the Zero Trust solution accordingly with the information and choices previously defined

5. Identifying Candidate Solutions: once we have made several proposals, we need to identify the right solution and maybe re-evaluate previous decisions and improve them
6. Initial Deployment and Monitoring: finally, we need to deploy the solution and all its components. In its early stages, we can start the solution in a sort of observation and monitoring mode, watching how it works and evaluating if our decisions were right or not. Just a few solutions do not present problems at the beginning, and we need to correct what does not work properly

Looking at State-of-the-art about Zero Trust, in the last years, all the major companies in the sector, like Google or VMware, has shown their idea of how it should be a Zero Trust Architecture.

Each proposed solution has its pros and cons, but if we remove the technological aspect, the underlying principles of continuous monitoring, authorization and authentication are quite the same. The following paragraph describes some of the most famous Zero Trust architecture deployed with three different approaches. The "problem" here is that all these solutions are developed for the IT world. This is because we have proposed a new approach to deal with cybersecurity in Industry, based on a ZTA idea, that cannot be considered entirely original in the IT field, but it is definitely for the OT. In the industrial sector, the State-of-the-art is still the I-DMZ, and only the most mature companies have started the transition to a Zero Trust approach.

### **1.4.2 Google BeyondCorp**

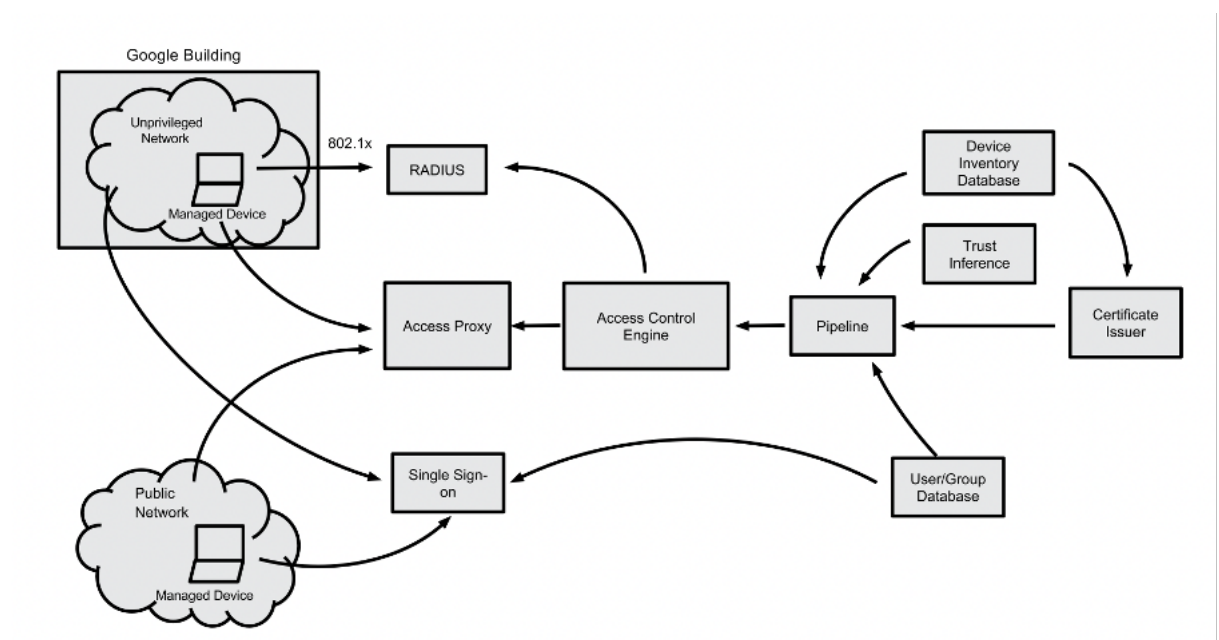
When we talk about technology and security, in particular, Google is the reference company, and BeyondCorp is the new Google Zero Trust approach to enterprise security. From many years Google has adopted the "open source" philosophy for the development of the majority of its own software and solution, and also, in this case, they share publicly the organization of their internal network infrastructure, something rare that make Google a unique case (as in many other fields).

The basic problem that made necessary this transition to ZTA is that perimeter-based security became problematic in a world in which remote work, cloud-based

technologies, and a huge variety of devices are increasingly widespread, a world where “Walls” does not work anymore. All the accesses to the infrastructure are granted based on device and user credential, and the level of trust is dynamically inferred from the specific context, who want to access? From where? with which device? Why? For how long? With this solution, there is no need for a Virtual Private Network (VPN) even in the case of a connection from a public network, and the access is uniform regardless of the location because the connection will be fully authenticated, authorized, and encrypted.

The main concept behind this new approach is to remove any idea of perimeter following the Zero Trust principle of “Never trust, always verify”. Everything inside or outside the network is untrusted until it is fully authenticated.

## Architecture



**Figure 1.3:** Beyond components and access flow.

- Device inventory Database:** As said, the authentication and authorization mechanism is based on two factors, user and device identity, these two information together determine not only if a specific user can access or not to the infrastructure but also the level of trust given to a specific user/device. Google uses the concept of “managed device” to replace trust in the network with an appropriate level of trust in the device. It is a device procured and actively managed by the enterprise, which ensures that the device is patched and does not present particular vulnerabilities.

It is essential because just the managed device can access the corporate network and application. Essentially the Device Inventory Database stores all the information needed for the tracking process of any devices of the organization.

Because of the huge number of devices managed by the organization, there is not just one Device Inventory DB, but there is also a central meta-inventory DB that aggregates all the data from all the Device Inventory Database (DIDB) to have an overall picture of all the devices.

One of the main challenges of this implementation is to overcome the heterogeneity of the devices that can access the network; in particular, Google differentiates the identification between desktops/laptops devices and mobile.

- Any device is uniquely identified by a certificate that is released only if the device is present and correct inside the DIDB. Once obtained, the certificate can be used in the identification process. The certificate is stored in hardware or software Trusted Platform Module (TPM) that is validated during the device qualification process. Only if the device is deemed sufficiently secure can be classified as a "managed device" and obtain full access to the company network.
  - In the case of a mobile device, a strong device identifier is natively provided by the mobile operating systems. For iOS devices, it is used the identifier ForVendor, while Android devices use the device ID reported by the Enterprise Mobility Management application.
- **User and Group Database:** The second factor for the identification is the user identity, the user/group database is used from a centralized Single Sign-on system, based on two-factor authentication that, after validating the credentials of the user that want to access to the company resources, release a short-lived token that plays a role in the authorization process for a specific resource. The User and Group database are continuously updated every time an employee is hired, leaves the company, or changes role to determine which resources and with which degree of privilege the specific user can access, following the Least Privilege principle.
  - **Unprivileged network:** To make more uniform local and remote access, Google has created an unprivileged network that resembles an external network but with a private address space. When an employee is inside

a Google building, he is assigned to this network that is separated from the other parts of Google's network from a strictly managed ACL (Access Control List).

- **Access Proxy:** A crucial role in the authentication and authorization process is done by the access proxy that handles coarse-grained company policies. In combination with the Access Control Engine, the access proxy is used not only as a common login point for all the external requests that arrive at the company but also to enforce and adapt policies to overcome the needs of the network more quickly. Another benefit of having a central component to handle all incoming requests is that this solution makes the forensics operation more effective when needed. The Access Proxy also provides other features like load balancing, access control checks, and denial-of-service protection and enforces encryption between the client and the application. After the access control checks, it redirects the request to the right back-end that can handle and enforce more fine-grained policies that the Access Proxy could not handle.

Within the Access Proxy, the Access Control Engine uses heuristics and all available information from several sources to infer the level of trust for a specific user and device.

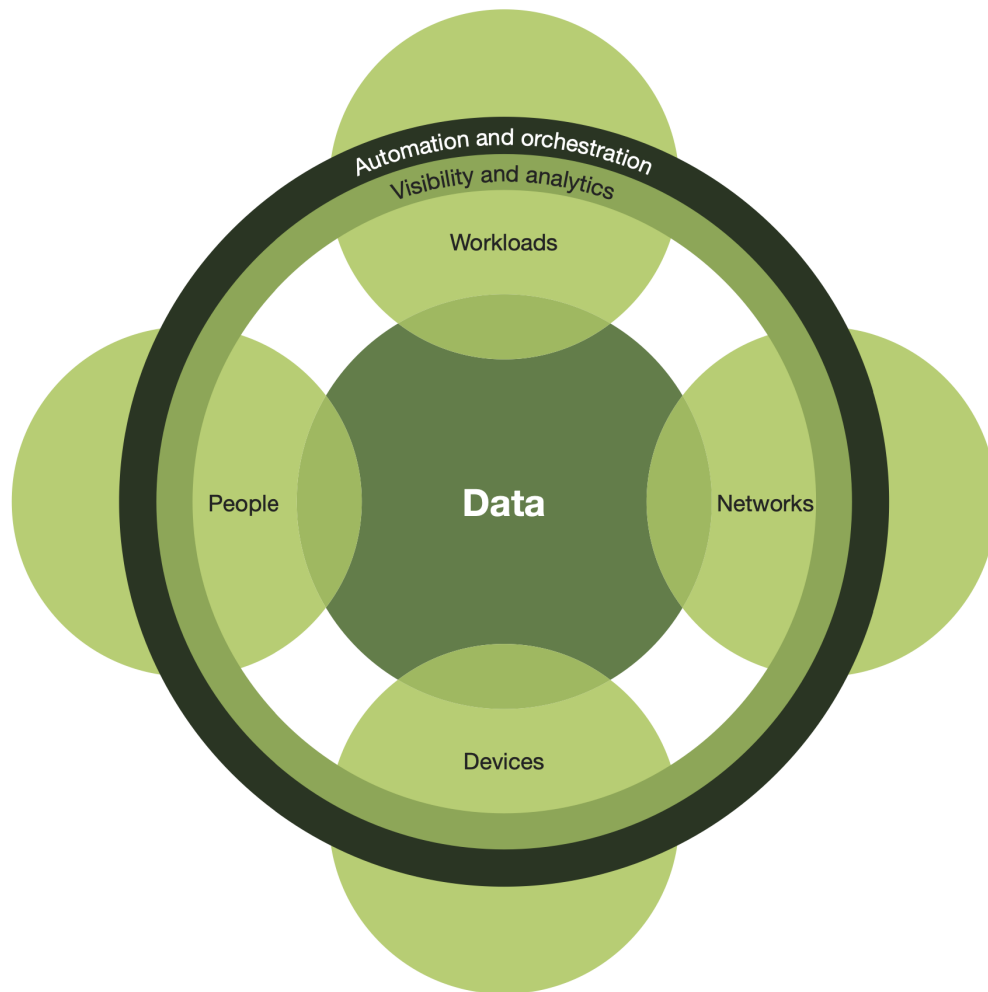
It is also fed constantly from a pipeline that dynamically extracts information from the user and device inventory database useful for access decisions.

### 1.4.3 Zero trust eXtended (ZTX) ecosystem Framework

Year after year, Zero trust has become a clear choice for any company that wants to build a scalable and effective security solution. Because of that, Forrester also considered it necessary to define its own Zero Trust Framework [11] to help companies during the transition from a perimeter-based security solution to a Zero Trust Architecture. The company has built what they call the "*Zero trust eXtended (ZTX) ecosystem Framework*", a control mapping framework for the evolution of a Zero trust ecosystem, to provide a reference point for any company and pros that want to implement this type of solution. Over time, during their research work, the company has realized that too often, the concept of Zero Trust was confused with network segmentation and the use of a Next-generation Firewall (NGFW).

Nevertheless, Zero Trust is more than network segmentation and, as we can see from the image 1.4, Forrester's Framework defines all the components needed for

a complete Zero Trust security solution.



**Figure 1.4:** Components of the Zero trust eXtended ecosystem.

- **Zero Trust data:** the Zero Trust model can be considered a data-centric model, and data security is one of the pillars. Data must be secured, managed, categorized, and encrypted at rest and in transit.
- **Zero Trust networks:** the ability to segment, isolate, and control the network continues to be a pivotal point of control for the Zero Trust approach. Many vendors have invested in realizing even more powerful and easy-to-use solutions to provide segmentation and isolation to their clients.
- **Zero Trust people:** As in any other security solution, the weakest link is users, because of that, a key pillar of a Zero Trust solution is limiting and strictly enforcing users' access. To do that, all the technologies for authenticating users and continuously monitoring and governing their access and privileges are needed.

- **Zero Trust workloads:** the workloads are all the front-end and back-end systems that run the business and help it win, serve, and retain customers. All the interactions between those components can be threat vectors and must have Zero trust controls and technologies applied.
- **Zero Trust device:** Zero Trust was born because the growing number of IoT and network-enabled devices have made the perimeter-based approach quite useless. All the "smart" things introduce a massive area of potential compromise for networks and companies. In order to move toward a Zero Trust strategy, security teams must be able to isolate, secure, and control every device on the network at all times.
- **Visibility and analytics:** *"you can't combat a threat you can't see or understand"*. tools like SIEM or security user behavior analytics (SUBA) are essential to control and monitor what happened inside your network, to empower the security analyst to accurately observe threats that are present and orient defenses more intelligently.
- **Automation and orchestration:** Forrester has done extensive research and analysis in this area and has shown just how critical it is for organizations and leadership to leverage and use tools and technologies that enable automation and orchestration across the enterprise.

The goal of Forrester's Framework is to help companies and professionals to build a Zero Trust "Ecosystem" with a holistic approach, in which all the components are interdependent, and the architecture evolves and develops in a continuous process. So it is important to provide a reference to determine precisely what tools and technologies are available and which companies should leverage for their security operations needs. However, as said, Zero Trust is a strategy, a mindset, it requires policies and procedures that only later are translated into specific technologies. This is because Forrester has built a control mapping framework that defines four steps for a correct evolution of the Zero trust ecosystem:

1. **Zero Trust strategy:** Define a strategy is the first point for an effective security. The team should understand the specific goals, and define a strategy to reach them, *"the strategy is to become a Zero trust ecosystem, not buy a Zero trust technology item and hope things are now "Zero trust."*

2. **Zero Trust capability:** A Zero Trust ecosystem is composed of several components with different capabilities. Before you even consider a specific technology or vendor is essential to define which are the capabilities, policies and procedures to reach goals and apply a strategy.
3. **Zero Trust technology:** Only after having defined your strategy and the capabilities required can you consider which technologies you need to build your ecosystem. the question that is needed to ask is *“What capabilities does this technology support and where does it specifically plug into my team’s Zero trust strategy?”*. Zero Trust is a constantly evolving process. This is because Forrester suggested choosing technologies compatible with heterogeneous solutions.
4. **Zero Trust feature:** This is the final and most granular step, we need to define which specific feature of a technology enables a capability to meet our Zero Trust strategy, and it is important that vendors can describe how the specific feature that they offer aligns with the other levels of the framework.

Forrester highlighted how the increased demand for Zero Trust solutions had led vendors and providers to work to improve their products, focusing on some specific aspects like:

- The incorporation of security, data, and business context
- Focus on ease of use
- Integration of functionality across products from different security domains
- Develop and support robust APIs and a partner ecosystem
- Maintain a center of gravity for visibility, analysis, policy, and automation

The idea of Ecosystem offered by Forrester perfectly sums up the Zero Trust approach and how companies can start the transition to this security model in a programmatic and effective way.

#### 1.4.4 VMware NSX Zero Trust Network

The previously described Forrester’s vision defines high-level practices and guidelines to build a Zero Trust solution regardless of specific vendor solution.

Also VMware has proposed its Zero Trust solution, but in this case, clearly, it was defined based on their products and technologies. Because of that, their Zero

Trust solution, exposed in the book "**Zero Trust Networks with VMware NSX**" [30] is much more "technical" and tailored over their specific technologies, trying to adapt them to the Zero Trust principles. This is also the reason why, reading the book, too often the concept of Zero Trust is used as synonymous with "Microsegmentation". Nevertheless, the VMware solution provides several interesting ideas for implementing a Zero trust approach. Looking at the NIST SP.800-207 publication defines several variations of Zero Trust Architecture Approaches, among all those presented, the VMware solution can be considered a Zero Trust using "Micro-Segmentation" based on a "Network Infrastructure and Software Defined Perimeters". A ZTA based on Micro-segmentation is organized, placing individual or groups of resources that share some common characteristics on a unique network segment protected by a gateway security component.

The VMware solution leverage two different technologies, Overlay network [12] or Software-defined Network (SDN) platform and Virtualization, to implement Micro-segmentation. An Overlay network is a logical network created on top of an existing physical one, enabling to virtualize the network layer by separating the control plane from the data plane. The idea is to virtualize infrastructure components like the firewall, router, load balancer, and VPNs into virtual functions, to create a "micro-services" architecture, in order to exploit the potentiality and flexibility of Virtual machines and containers.

This solution also allows to modify as little as possible the underlying infrastructure, reducing the time of implementation and the "fear" derived from the modification of the infrastructure.

The NSX platform provides a REST API interface to virtualize and configure all the components of a network like:

- **VXLAN:** this is a tunneling protocol used to simulate the behavior of a VLAN. In this way, we can virtualize the segmentation of the network. It uses an additional header to encapsulate the packets that, in this way, can travels through the network underlay as a normal IP packet. After reaching the destination, the packet will be decapsulated and sent to the correct virtual Network Interface Card (vNIC).
- **Distributed Router:** A distributed logical router is a virtual component of NSX used to prevent a considerable amount of traffic from flowing to the physical router.
- **NSX Distributed Firewall (DFW):** Also in this case, we are talking of a virtualized firewall, in particular, DFW simulates the behavior of a stateful



firewall keeping track of the connections and acting following its previous decision.

With DFW, we can define per vNIC rules, and this means that each packet originating from a VM has to pass through the filtering module before the packet even reaches outside the ESXi<sup>2</sup>. This helps block the traffic in the virtual machine space itself.

With this software-defined solution, we can automate the entire infrastructure creation process, making it easier its configuration and maintenance. The transition to a Zero Trust approach is made more accessible by virtualization and the possibility of creating this Overlay network to manage all the aspects of the infrastructure, from the firewall rules to the segmentation via software commands, without the need for a "physical" action.

This solution provided by VMware is undoubtedly valid and robust. It allows us to define policies and procedures quickly, making it easier to modify the architecture and the security posture according to the moment's needs, defying firewall rules, or segmenting the network on the fly.

As said, the idea behind this solution is very effective and powerful, the problem is that it requires to rely totally on a vendor-defined solution, which can create a lock-in problem.

---

<sup>2</sup>VMware ESXi is the bare-metal hypervisor in the VMware vSphere virtualization platform. As a bare-metal hypervisor for creating and running virtual machines (VMs), VMware ESXi runs on top and directly accesses the hardware without needing to install an operating system

# ICS cyber-attacks

From many years ICS has become one of the main targets of APT (Advanced Persistent Threat) or attackers in general. This is because attacks to these types of systems can have a significant impact, causing not only loss of money but also damage, degradation, and potentially the destruction of industrial environments and physical processes. In the following paragraph will be described the main frameworks that help companies and pros to understand which are the Techniques, Tactics, and Procedures (TTP) used by the attacker to intrude Industrial infrastructure.

These tools are fundamental for many reasons:

- First of all, they provide a common language, making it easier to share knowledge and new information among the experts in the sector
- They are helpful during the risk assessment phase to identify threats and vulnerabilities that affect our systems, weighting the potential damage, and prioritize problems.
- The focus of these frameworks, especially for the MITRE, is to highlight which are the TTP used by an attacker, to better define an effective security posture, regardless of the specific technology we use.
- they also provide a lot of intelligence information and a model over which AI technologies can be developed

## 2.1 Security by obscurity

The lack of security in industrial environment has historical reasons. Until recently, before we started to talk about Industry 4.0 and IIoT, the industry and in particular OT network security relied on the concept of ‘Security by obscurity’. The network was physically isolated from the rest of the infrastructure and

especially from the internet, they used proprietary protocol that does not have any idea of security (Modbus) and the only security they had to worry about was physical security.

With the advent of the last industrial revolution, companies have started to connect the plant to IT or, even worst, to the Internet, for several reasons, from the capacity to control the implant remotely, up to the possibility of processing data to extract useful information and value form them.

Being connected became rapidly an unavoidable requirement to remain competitive in the market.

The problem, at this point, is that this revolution did not bring any awareness of the risks associated, leaving companies at the mercy of cybercriminals.

## 2.2 MITRE ATT&CK for ICS

MITRE<sup>1</sup> ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) for ICS can be considered the first consolidated and centralized encyclopedia of know Tactics, Techniques, and Procedures for Industrial systems that summarizes observed behaviors of industrial attackers.

It is one of the three modules of MITRE ATT&CK, the other two are for enterprises mainly focused on adversarial behavior in Windows, Mac, Linux, and Cloud environments and for Mobile that is about describing actions of attackers in IOS or Android operating system.

This massive collection of information is the result of a considerable effort from the MITRE and community contributors that every day undertake to identify new threats, analyzing reports about real-life attacks discovered in the ICS environment.

One of the best results of the MITRE ATT&CK framework project is the MITRE ATTCK Matrix 2.1, it is updated with all the known TTP (Tactics, Techniques, and Procedures) used from attackers.

---

<sup>1</sup>MITRE is an American non-profit organization supporting various U.S. government agencies in the aviation, defense, healthcare, homeland security and cybersecurity field.

**ICS Matrix**

Below are the tactics and techniques representing the MITRE ATT&CK® Matrix for ICS.

Initial Access	Execution	Persistence	Privilege Escalation	Evasion	Discovery	Lateral Movement	Collection	Command and Control	Inhibit Response Function	Impair Process Control	Impact
12 techniques	9 techniques	5 techniques	2 techniques	6 techniques	5 techniques	6 techniques	10 techniques	3 techniques	13 techniques	5 techniques	12 techniques
Drive-by Compromise	Change Operating Mode	Modify Program	Exploitation for Privilege Escalation	Change Operating Mode	Network Connection Enumeration	Default Credentials	Automated Collection	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	Damage to Property
Exploit Public-Facing Application	Command-Line Interface	Module Firmware	Hooking	Exploitation for Evasion	Network Sniffing	Exploitation of Remote Services	Data from Information Repositories	Connection Proxy	Alarm Suppression	Modify Parameter	Denial of Control
Exploitation of Remote Services	Execution through API	Project File Infection		Indicator Removal on Host	Remote System Discovery	Lateral Tool Transfer	Detect Operating Mode	Standard Application Layer Protocol	Block Command Message	Module Firmware	Denial of View
External Remote Services	Graphical User Interface	System Firmware		Masquerading	Remote System Information Discovery	Program Download	I/O Image		Block Reporting Message	Spoof Reporting Message	Loss of Availability
Internet Accessible Device	Hooking	Valid Accounts		Rootkit	Wireless Sniffing	Remote Services	Man in the Middle		Block Serial COM	Unauthorized Command Message	Loss of Control
Remote Services	Modify Controller Tasking			Spoof Reporting Message		Valid Accounts	Monitor Process State		Data Destruction		Loss of Productivity and Revenue
Replication Through Removable Media	Native API						Point & Tag Identification		Denial of Service		Loss of Protection
Rogue Master	Scripting						Program Upload		Device Restart/Shutdown		Loss of Safety
Spearphishing Attachment	User Execution						Screen Capture		Manipulate I/O Image		Loss of View
Supply Chain Compromise							Wireless Sniffing		Modify Alarm Settings		Manipulation of Control
Transient Cyber Asset									Rootkit		Manipulation of View
Wireless Compromise									Service Stop		Theft of Operational Information
									System Firmware		

**Figure 2.1:** MITRE ICS TTP matrix.

It starts from the ‘initial access’ to cover the entire process for a successful attack, with this tool, a defender can map their detection mechanism to be able to more efficiently and consistently anticipate and counter threats.

Independently from the environment IT or OT, the historical goal of defenders is to identify which are network anomalies or IoCs (indicators of compromise) that would allow them to identify cyberattacks.

The problem is that at the same time, adversary rapidly changes their tools to evade these detection techniques with the result that attackers are always one step ahead to the defender.

This cycle of continual evasion made it necessary to change the approach of researcher and defender going beyond the technical detail and tools used by the attacker to focus their efforts on understanding which are the TTP, the behaviors of their adversary.

While IoCs or anomalies change rapidly, weekly or even daily, TTP change annually, this gives a great advantage to defenders that can adapt and seek more persistent and trasponibile forms of threat detection.

Using tactics, techniques, and procedures as tools to defend environments improve dramatically the security of systems and the defender’s ability to protect against adversaries. However, at the same time, it requires a lot more effort to set the defenses.

Focusing on tactics, techniques, and procedures forces to define a strategy for cyber defense, we cannot only set firewalls against a specific vulnerability or threats, but we need a well-defined strategy, and technologies with capabilities of threat intelligence (easily found in cloud solutions) and threat behavior analysis like SIEM or NGFW.

This is the reason why in 2013 is born the MITRE ATT&CK framework, while the Enterprise one is focused on the IT part, the ICS version was specified and created to help those who work in the ICS environment.

Before this framework, researchers and defenders had to analyze personally, and with a significant effort, all the public and non-public reports to understand how to set their defenses, MITRE ATT&CK create a standard for ICS/OT adversary knowledge, making much easier the life of defenders.

## 2.3 ICS Cyber Kill Chain

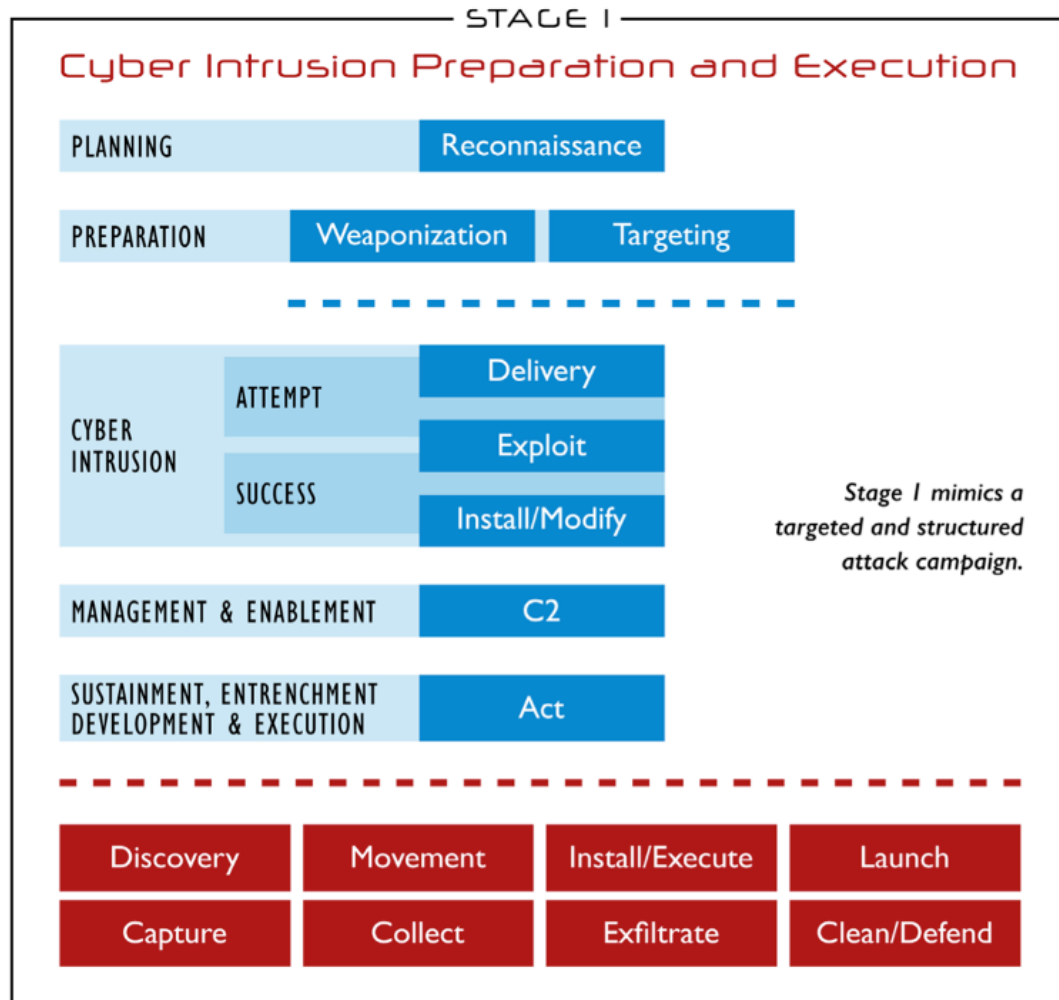
Another important framework for understanding adversary behavior is the ICS Cyber Kill Chain.

It takes inspiration from Lockheed Martin's Cyber Kill Chain [67], a security framework created in 2011 by Lockheed Martin analysts Eric M. Hutchins, Michael J. Cloppert and Rohan M. Amin to help in the detection and response of cyber attacks to IT environment, that was itself an adaptation of the military concept of the kill chain.

ICS environment is different from IT, and the original Cyber Kill Chain cannot be applied to it, so it was necessary to create one specific.

It is composed of two stages, in turn, composed of several steps through which an attacker must pass to conduct a cyberattack on an ICS successfully.

### 2.3.1 Stage 1



**Figure 2.2:** Stage 1 Cyber kill chain.

As we can see from the image 2.2, this first stage is characterized by all those operations that can be included in the category of espionage or intelligence to acquire all possible information about the target ICS.

The goal is to understand the system and defense behavior to implement a mechanism to defeat internal protection and intrude in the environment.

#### Planning

The first phase of the first stage is the planning phase, also called Reconnaissance, where adversaries conduct research through information-gathering services like Google (using google Dorks), Shodan, or Social networks.

In this phase, attackers try to gather the highest amount of information about

the target company, and those data can include any information like processes, procedures, system vulnerabilities, policies, human behavior, or network that can be useful during the attack.

This process takes advantage of the countless amount of information publicly disclosed online.

The Reconnaissance phase is crucial for the subsequent steps and is also the most difficult to detect as it happens. However, if defenders manage to identify the attack at this stage they can gain a great advantage over the attacker, maybe even understanding which is the intent and the specific target.

To do that, defenders can collect website visitors, build specific detections for browser behavior activities, or harden the defense of a particular device or service, based on the acquired information, to do that threat intelligence or SIEM technologies.

### Preparation

This second phase of the first stage consists of the weaponization of the target and can be considered the first operational phase of the attack.

The adversary starts to modify files, PDFs or documents to embed an exploit or to use maliciously normal features using the *"Living off the Land"* approach [22]. Typical, in this case, is to use maliciously the programs of the Office suite, widely used in practically every company that became one of the most used vectors to gain a foothold into a network.

This phase can also include all the procedures used by the attackers to define the best path to conduct the attack, also called **'targeting'**, defining which are the best tools to use and the appropriate actions to create the desired effect.

Cybercriminals are opportunists; they want to achieve the best result with the least effort possible. In this part of the attack, they make all the considerations about the costs and effects of the attack defining priorities among the different targets.

### Cyber intrusion

This phase is composed of two steps.

- **Delivery step:** in this step, the adversary starts to send the malicious files weaponized in the previous phase using, for example, phishing emails as vectors.

- **Exploit step:** there is where the attacker uses the files to initiate the attack, trying to install some tool to gain remote access into the network. In this phase, as in the others, attackers tend to prefer using the system's existing capabilities in a malicious way. This is because, in this way, they can reduce the cost of the attack and, at the same time, make it harder to detect their actions.

### Cyber intrusion

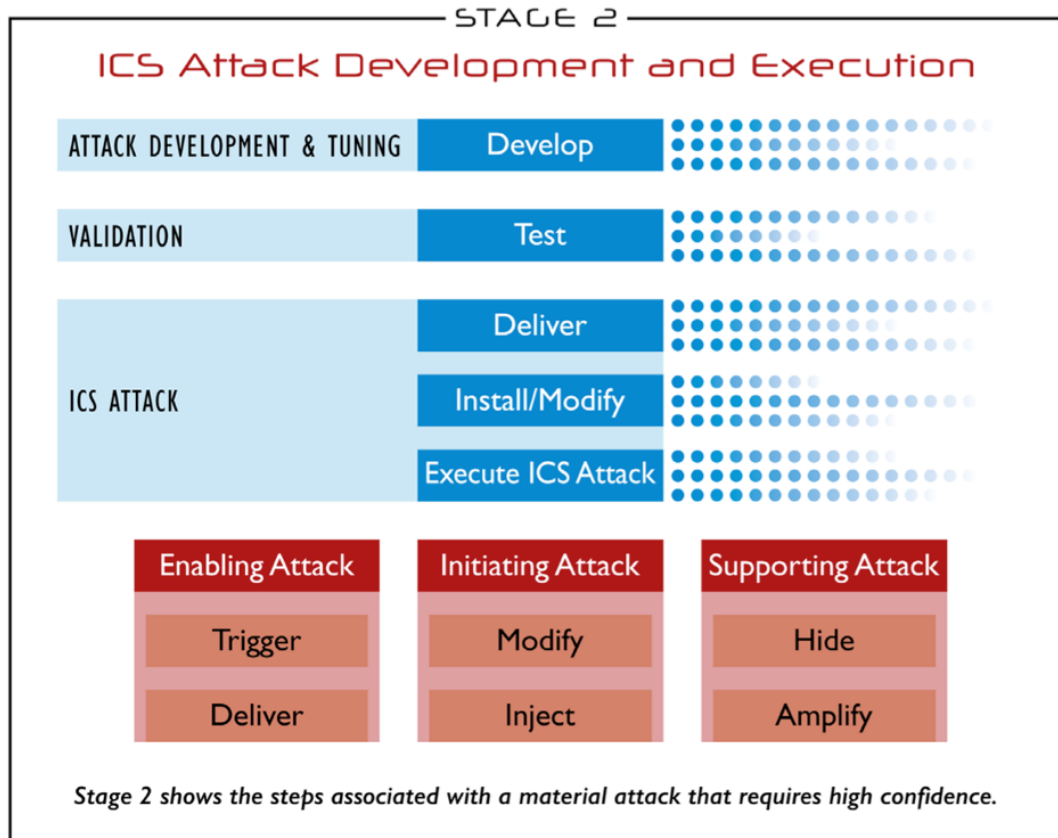
This is the last phase of the first stage, in which the attacker establishes C2 (command and control) capabilities using what have installed in the previous steps.

The attacker is now inside the network and can start to act to pursue their goals. In an ICS attack, this phase has the goal of intruding into the corporate network of the company and gathering all the information needed to intrude into the Control network, that is the typical goal.

This phase gains much more importance if the attacker's goal is to exfiltrate information useful for intelligence or espionage in case of nation-sponsored attacks, or over which they can monetize in case of cybercriminals.



### 2.3.2 Stage 2



**Figure 2.3:** Stage 2 Cyber kill chain.

In this second stage, the adversary leverage all the information gathered in Stage 1 to attack the ICS, this does not mean that the first stage cannot cause damage to the infrastructure. All the operations of the first stage require, in fact, interaction with the ICS that could bring to an unintended attack

It is also important to highlight that the delay between the two stages can also be significant. The attacker can stay hidden in the network for months, during which it exfiltrates more and more information. This second stage is the one in which the intentional attack start, it is composed of three main phases.

- **Attack Development:** This is the phase during which all the capabilities are developed to attack the victim based on the information acquired and tailor the tools and TTPs to the specific ICS implementation.
- **Validation:** in this part of the kill chain, the attacker test what he has created until now to define if it really works. A solution could be to buy the physical or software equipment and test the attack.

This is one of the most important differences between IT and ICS cyber-attacks. While IT cyberattacks (typically) do not require a great number of resources, an ICS attack is much more complicated and costs more both in terms of effort and resources. It requires a really competent team with a huge amount of resources, and this is also because ICS cyberattacks are often sponsored by nation actors.

- **ICS Attack:** this is the time of the real attack, this may be composed of several parts. An important part of the attack could be composed of all the actions and procedures required to support itself. For example, it may require time to be completed, which means that the attacker must stay hidden and let operators think that everything is okay.

The ICS Cyber Kill Chain and the MITRE ATTCK for ICS can be considered the Industry standard and are the most used frameworks for understanding and communicating how attacks work.

## 2.4 Legacy systems

Connected with the idea of ‘Security by obscurity’, there is the other huge problem for the security of Industrial architecture, which is legacy systems.

Different from IT, where a typical life cycle of a system is a couple of years, this is not true in OT. In these types of environments, the system could stay up also for decades because the idea is that until something works, it is fine.

Even if this is a limit for innovation, it becomes a massive problem for the security of systems that are often unpatched or with old software installed, putting the entire architecture at risk.

Indeed, legacy systems can be considered as a vulnerability for the infrastructure. They are no longer supported by the vendor and does not receive patch or update for the hundreds of vulnerabilities that are discovered every day.

Nevertheless, today we can find industrial infrastructure, even the most critical, relying on systems based on very old software, and manager or administrator does not want to change them.

In this case, the only thing we can do as engineers is trying to mitigate the associated risks as much as possible.

The process used to secure these types of systems is called hardening, which can be described as the set of tools, methods, and recommended procedures used to reduce the attack surface in the technological infrastructure, including software,

data systems, and hardware.

System Hardening aims to reduce the overall "threat profile" or vulnerable areas of the system, looking at which are the vulnerabilities that affect it through a risk assessment specific to it, and, once identified all the possible vulnerabilities, pass to the risk management phase. Unfortunately, it is not rare to find a legacy system in critical machines like HMI or Engineering workstations, devices like this are typically designed for a specific machinery that can be used for several years, often decades, an eternity for the world of security. Other types of devices like the Domain Controller or Data Historian suffer less of this problem, and this is because they can be considered more "general" devices, not associated with a specific machinery, that only need to be backward-compatible.

## 2.5 Famous attacks

In the following sections will be reported the most famous attacks on Industrial Control Systems. Over the years, these could evolve significantly faster than the security in ICS, managing to significantly compromise the production capacity, the security, and the safety of companies and Critical infrastructures.

### 2.5.1 Stuxnet

When we discuss about malwares and cyber attacks on Industrial networks, it is impossible not to talk about Stuxnet [34], considered the first known cyberweapon. It was identified and reported for the first time in 2010, even if experts believe that its development started in 2005. the worm managed to infect more than 20,000 devices in 14 Iranian nuclear facilities and ruined around 900 centrifuges through several "zero-day" vulnerabilities.

It was a very "smart" malware, able to pass across air-gapped systems through a USB stick. It was also designed to activate only if in the machine is installed the Step 7 software from Siemens.

Once injected through a USB device into a device directly connected to the power plant, it slowed down and speed up secretly the engines of the Iranian centrifuges that were used to enrich uranium, as well as valves that, in turn, connect six cascades of centrifuges. It also has a code for a man-in-the-middle attack that fakes industrial process control sensor signals so an infected system does not shut down due to detect abnormal behavior. From Stuxnet derived a lot of other malwares that were based on its original code.

### 2.5.2 Industroyer

Industroyer is one of the "sons" of Stuxnet, also known as "Crashoverride", and is thought to have been used in the cyberattack on Ukraine's power grid on December 17, 2016. It is a sophisticated piece of malware [28] designed to cause an Impact on the working processes of Industrial Control Systems (ICS). It is the first malware ever seen to have been specifically designed to attack power grids. During the years, several variants of the original malware have been developed, and in particular, as reported from Mandiant [38], on April 12, 2022, a cyber-physical attack impacted operational technology (OT) supporting power grid operations in Ukraine. The attack leveraged different pieces of malware, including a variant of Industroyer. For this reason, the malware was called Industroyer V2, this means that despite five years of substantial analysis into Industroyer from a variety of researchers, the actor still attempted to repurpose the tool and customize it to reach new targets.

### 2.5.3 Triton

Another famous malware that is considered a "son" of Stuxnet is Triton, also considered the world's most murderous malware [39] because it can disable physical safety systems designed to prevent catastrophic industrial accidents.

It is known for the attack against a Middle East-based petrochemical facility's safety controllers during the summer of 2017. In particular, the malware targeted the Schneider Electric Triconex safety instrumented system (SIS), which is used to initiate safe shutdown procedures in the event of an emergency. The malware also gives the attackers complete remote control of the SIS, providing them the capability to cause significant physical damage and loss of life if the plant were to enter an unsafe state. Schneider Electric addressed the vulnerability when version 11.3 of the Tricon controller was released in June 2018.

However, older versions of the controller remain in use and are vulnerable to a similar attack, this is because a recent FBI report [27] highlights how dangerous the malware is, recommending the companies to regularly assess and monitor their SIS systems, personnel with access to these systems, and practice contingency plans, maintain updated their systems in order to reduce further the risks of these type of malware attacks.

## 2.5.4 Pipedream/Incontroller

Pipedream/Incontroller is the seventh-known Industrial Control Systems (ICS)-specific malware after Stuxnet, Havex, Blackenergy2, Crashoverride/Industroyer, Trisis/Triton, and Industroyer2, developed by the CHERNOVITE group, to manipulate and disrupt industrial processes.

It is extensible and composed by several tools that allow to conduct a complete attack from the reconnaissance phase to the exploit and command and control, covering the ICS Kill Chain Stages 1 and 2, and 36 out of 88 of the MITRE ATTCK for ICS technique as we can see from the image 2.4.

INITIAL ACCESS	EXECUTION	PERSISTENCE	PRIVILEGE ESCALATION	EVASION	DISCOVERY	LATERAL MOVEMENT	COLLECTION	COMMAND & CONTROL	INHIBIT RESPONSE FUNCTION	IMPAIR PROCESS CONTROL	IMPACT
Data Historian Compromise	Change Operating Mode	Modify Program	Exploitation for Privilege Escalation	Change Operating Mode	Network Connection Enumeration	Default Credentials	Automated Collection	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	Damage to Property
Drive-by Compromise	Command-Line Interface	Module Firmware	Hooking	Exploitation for Evasion	Network Sniffing	Exploitation of Remote Services	Data from Information Repositories	Connection Proxy	Alarm Suppression	Modify Parameter	Denial of Control
Engineering Workstation Compromise	Execution through API	Project File Information		Indicator Removal on Host	Remote System Discovery	Lateral Tool Transfer	Detect Operating Mode	Standard Application Layer Protocol	Block Command Message	Modify Firmware	Denial of View
Exploit Public-Facing Application	Graphical User Interface	System Firmware		Masquerading	Remote System Information Discovery	Program Download	I/O Image		Block Reporting Message	Spoof Reporting Message	Loss of Availability
Exploitation of Remote Services	Hooking	Valid Accounts		Rootkit	Wireless Sniffing	Remote Services	Man in the Middle		Block Serial COM	Unauthorized Command Message	Loss of Control
Internet Accessible Device	Modify Controller Tasking			Spoof Reporting Message		Valid Accounts	Monitor Process State		Data Destruction		Loss of Productivity and Revenue
Remote Services	Native API						Point & Tag Identification		Denial of Service		Loss of Protection
Replication via Removable Media	Scripting						Program Upload		Device Restart/Shutdown		Loss of Safety
Rogue Master	User Execution						Screen Capture		Manipulate I/O Image		Loss of View
Spearphishing Attachment							Wireless Sniffing		Modify Alarm Settings		Manipulation of Control
Supply Chain Compromise									Rootkit		Manipulation of View
Wireless Compromise									Service Stop		Theft of Operational Information
									System Firmware		

Figure 2.4: Pipedream/Incontroller TTPs.

Being its discovery very recent, it is important to underline that there have been two independent analyses, one from Dragos [16] and an other from Mandiant [37], both led to the same results but in the former the malware is called Pipedream while in the latter the name is Incontroller. As other malware of this type Pipedream can be considered as a toolkit of several components as identified by Dragos whitepaper.

These different module can be used together or separately to achieve different goals.

## EVILSCHOLAR

A tool capable of interact, discover, manipulate and disable Schneider Electric PLCs using CODESYS and Modbus libraries.

- Developed in Python and Linux ELF (Executable and Linkable Format)
- Run a scan that identifies all Schneider PLCs on local network from a device already compromised via UDP
- Brute force Schneider PLCs passwords
- Conduct CODESEYS DoS (Denial-of-service) attack to interrupt communication with PLCs
- Logging in/out, uploading/downloading files

An important functionality of Evilscholar is the so-called PLC Proxy. Supposing that we have a host/workstation used specifically to communicate with a PLC that is behind a Firewall.

Once that the attacker compromise a workstation, for example in the DMZ, the tool can use it as a pivot to attack the PLC, bypass the firewall and gain the control of the PLC.

Dragos has identified three steps of the PLC PROXY module:

1. From the compromised Engineering workstation we can conduct an enumeration to identify the specific PLC's gateway in the OT network.
2. Once that it is find we can add the Route to the compromised workstation to enable proxy communication through the simple command: **\$ip route add gateway\_ip 24 dev nic via plc\_ip**
3. Once established the route, Evilscholar can send Modbus commands to the PLC.

It can also Enumerate other devices in the OT sending a *Modbus Read Device information request* to the PLC and then the PLC, because of a feature of Schneider device, will send this request to the internal network.

The problem there is that with this functionality once that we have the control of a PLC, even if the Engineering workstation that we have compromised should communicate with just on PLC, we can proxy other PLC through the compromised one.

## **BADOMEN**

This module provide remote shell capability, it is similar to Evilscholar in terms of functionalities except that is designed to interact with Omron software and PLCs. It can use several methods to log into the PLC but, even more dangerous, it is capable to remotely turn on telnet and then use it to upload an implant into the PLC.

It can also upload executable, binary or files in general to the PLC, terminate PLC connections or deploy Tcpdump to analyze network traffic.

The tool has also other powerful capabilities like:

- Manipulate Servos via EtherCat
- Change Operating mode of the PLCs
- Wipe the PLC's memory

Thanks to these capabilities this module is really dangerous and able to cause serious damages to the industrial plants and put at risk the safety of peoples.

## **MOUSEHOLE**

A multi-platform toolkit for interacting with OPC-UA servers written in Python. It is designed to read and write node attribute data, enumerate the Server Namespace and associated NodeIds. It can brute force credentials of OPC UA servers using a list provided by the user and it can also read the server structure and modify some attributes. It can be used to a command and control attack to a OPC UA server establishing a foothold in the network and then continue the attack.

## **DUSTTUNNEL**

It has custom remote operational implant capability to perform host reconnaissance and command-and-control on Windows, it is a C++ compiled binary, and essentially give a remote command prompt to the host machine.

It has also the capabilities to:

- Enumerate victim host machine

- Enumerate network connections.
- Execute, install or delete modules
- Upload/download files

It use anti-debugging and anti-analysis techniques to remain undetected.

## LAZYCARGO

The last module is Lazycargo, a Windows executable that has the ability to drops and exploits a vulnerable ASRock motherboard's (CVE-2020-15368) driver to load an unsigned driver.

It requires administrator access to install the vulnerable driver as described in the repository [69], it provides the capability to execute malicious code in the kernel of Windows machine.

The problem is that these types of vulnerable motherboard are used in HMI or Engineering workstation making dangerous this module for OT security.

A first analysis of the malware shows that it was developed to compromise specific ICS/SCADA device of two vendors, Schneider Electric and Omron, in particular as we can read from the CISA report [5]

- Schneider Electric MODICON and MODICON Nano PLCs, including (but may not be limited to) TM251, TM241, M258, M238, LMC058, and LMC078;
- OMRON Sysmac NJ and NX PLCs, including (but may not be limited to) NEX NX1P2, NX- SL3300, NX-ECC203, NJ501-1300, S8VK, and R88D-1SN10F-ECT; and

But it was immediately clear to experts that also infrastructure that doesn't use these technologies must be aware of the risks and implement the necessary security measures to protect their infrastructure from this malware.

The expert advice to focus mainly on tactics and techniques used by the malware rather on the specific module's functionalities that can change rapidly.

Because of the modular nature of this malware, developer can easily create new modules extending the capabilities of the malware and the number of vulnerable devices.

This capability to be extended, make it a big deal for ICS infrastructure and shows that developers intend to support the tool long term and adapting it to the new operational requirements.

Comparing it with common red team tools such as Metasploit and PowerShell



Empire, Pipedream is quite easy to use, for experts this is also a sign that the developers of the malware knew that the malware would be used also from not skilled peoples.

Pipedream is also capable to leverage vulnerabilities of several ICS Protocols, this means that it can potentially affect device of any vendor especially considering how widespread some of these protocols are:

- **Codesys:** Codesys <sup>2</sup> stands for ‘Controller Development System’ and is an IEC 61131-3 programming tool, it is used in automation and is a platform independent environment compatible with PLC and automation components from many different companies.

Over 1000 different types of devices are compatible with Codesys, this means that a malware capable of interact using this protocol can be used to compromise a huge number of infrastructures.

It support all common processors and operating system, manufacturers use a toolkit to port the CODESYS runtime system on their device and turn it into a CODESYS-compatible IEC 61131-3 PLC.

- **Modbus:** The Modbus <sup>3</sup> protocol was developed in its first version ‘Modbus RTU’ in the 1979 by Modicon (now Schneider Electric) as a proprietary protocol for their own industrial automation systems.

The protocol is becoming a standard for communications in the Industrial world thanks to its efficiency and adaptability to any embedded device. It works to level seven of the OSI model [72] and has a master/slave architecture with a request/response model of interaction in which slaves can only send a response to the master.

The second most spread version is Modbus TCP/IP that is simply the Modbus RTU protocol with the TCP interface on Ethernet.

The protocol defines a messaging structure, the rules to organize and interpreting the data independently from the communication medium, the TCP extension use the protocol to carry the data from a device to another.

The protocol owes its popularity to its simplicity but, for the same reason, it is very unsecure and presents several vulnerabilities particularly easy to exploit. It lacks in:

- Confidentiality: there isn’t any encryption methods

---

<sup>2</sup>Codesys. url: <https://www.codesys.com>.

<sup>3</sup>Modicon. Modbus protocol. url: <https://www.modbustools.com/modbus.html>

- Integrity: there isn't any integrity check in the protocol which therefore must rely on the protocol of lower levels
- Authentication: any authentication phase is expected

There are a lot of other weakness of the protocol that for many years did not pose any problem for the industries security because, as said, they rely on 'security by obscurity', but now this is no longer true and the protocol has become an important problem for the OT security.

- **OPC-UA:** OPC<sup>4</sup> (Open Platform Communications) is a communication standard for industry 4.0 and IoT created from the OPC Foundation, a no-profit organization, it has the goal to standardize the access to machine and industrial devices.

Its successor is the OPC-UA (Unified Architecture) was published in 2008 and has the aim to be platform independent, allowing interoperability between devices independently from the proprietary APIs of manufacturers.

It made easier the communication between HMI, PLCs and other industrial devices that typically producing data each with its own format. Along with Modbus it can be considered as the most important protocol in manufacturing industry.

It is based on a client/server architecture and uses TCP/IP and HTTP/-SOAP (OPC was using COM/DCOM protocol).

Differently from Modbus, OPC UA integrate some security functionality like PKI certificates, Web Socket tokens and authentication processes for clients

### General attack description

As many others attack to Industrial plants, a possible attack start in the IT after a phishing campaign or a compromission of a machine (BadUSB) that allows to install the DUSTTUNNEL module in the compromised device that give a remote command prompt to the compromised machine.

Once that we have taken the control of a machine, with DUSTTUNNEL, we can install other software to continue our attack.

A possibility is, for example, to use Mimikatz to gather credentials to access a legitimate account and gain a persistent foothold in the enterprise network.

Thanks to enumeration capabilities of DUSTTUNNEL we can determine the location of other machines and use the captured credential to move in the network,

---

<sup>4</sup>opcfoundation. OPC-UA protocol. url: <https://opcfoundation.org/about/opc-technologies/opc-ua/>

intruding for example DMZ machines.

At this point, once that the attacker is inside the network, if there is a OPC UA server, the attacker could use MOUSEHOLE to gain the control of the server, brute force the credential and use it to interact and send malicious command to PLCs or other OT machines.

If the attacker has gain also the control of a machine in the I-DMZ and there is an HMI or Engineering workstation he may use LAZYCARGO to install the malicious device driver and essentially take the control of the communication and consequently of the PLCs.

Another option is also to use EVILSCHOLAR, once that the attacker control a machine in the OT, to interact with Schneider Electric PLCs causing damages to the industrial machines which may also put at risk the safety of peoples.

When we talk about Schneider Electric or Omron PLC must be clear that this doesn't means that this specific devices has some particular vulnerabilities, this malware is very flexible and developed to be used globally.

What experts have found is that the 'initial' configuration is targeting devices from these vendors (Shneider Electric and Omron) but is clear that it can be used to attack different types of devices thanks to its capabilities. The Figure 2.5 shows how Pipedream and its modules can be mapped over an Industrial plant

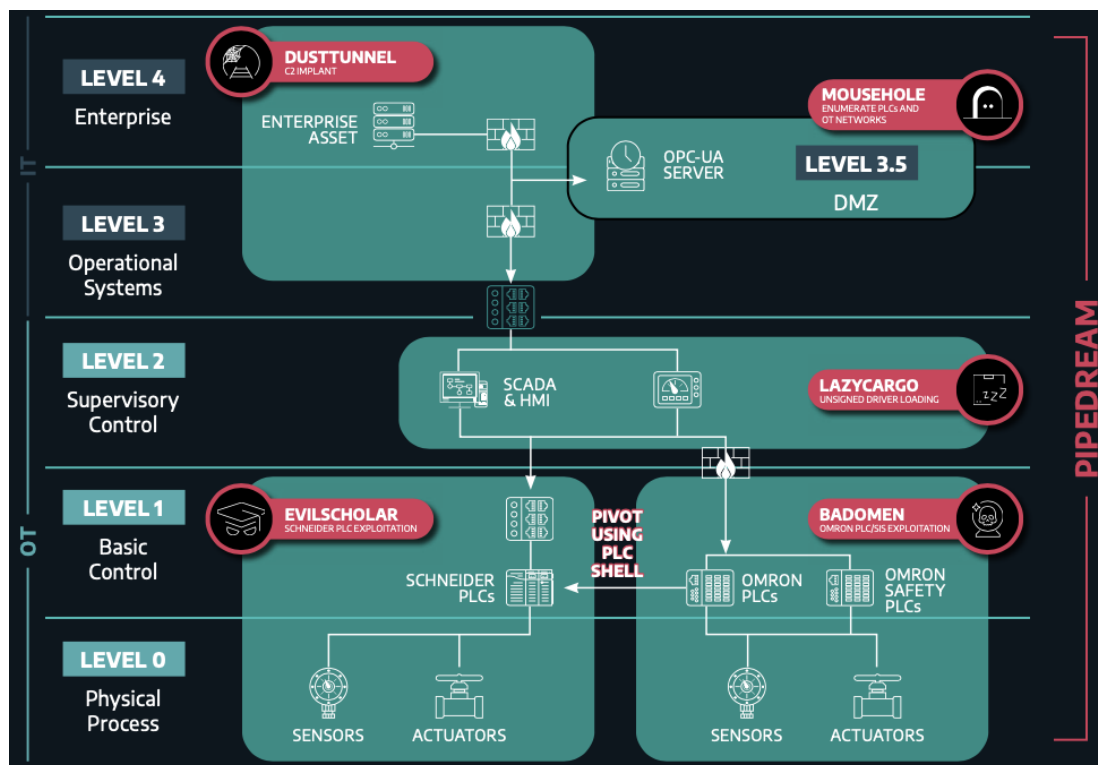


Figure 2.5: Pipedream/Incontroller modules.

# Proposed solutions

In this chapter, we propose and describe three different approaches for industrial cybersecurity. The first one uses the I-DMZ framework, we propose some innovative solutions to mitigate some problems of this approach. The second architecture is the core of this thesis and it is the real innovative proposal of this work based on Zero Trust. Finally, the third architecture shows how Cloud can make the implementation of Zero Trust easier and how Cloud providers offer several services that can be considered Zero Trust oriented.

## 3.1 Industrial Demilitarized Zone

For the I-DMZ, our solution, showed in Figure 3.1, is composed of a network divided into three subnets. The first one is the IT, where are placed all the generic services like the:

- VPN server (for remote connections).
- Enterprise resource planning (ERP) <sup>1</sup>, Mail server and other generic servers.
- IT Domain controller (for the authentication and authorization).

In the OT are placed:

- The Engineering workstation <sup>2</sup>
- The HMI (Human-machine Interface)<sup>3</sup>.

---

<sup>1</sup>ERP is a software system that helps companies to run their entire business, supporting automation and processes in finance, human resources, manufacturing, supply chain, services, procurement, and more.

<sup>2</sup>The engineering workstation is usually a high-end, very reliable computing platform designed for configuration, maintenance, and diagnostics of the control system applications and other control system equipment

<sup>3</sup>It can be a device or a software used to communicate with PLC, machinery and production plant

- Data Historian <sup>4</sup>
- OT Domain controller
- PLCs and sets of sensors and machinery.

Finally, between IT and OT, there is the I-DMZ.

The goal of an I-DMZ is to avoid any direct connection from the corporate network to the control network to take advantage of the “defense-in-depth” principle [24]. From the implementation point of view, in the DMZ are placed all the assets that could be contacted both from the IT and the OT network.

The fundamental components of our I-DMZ, also showed in Figure 3.1 are:

- Data Historian mirror
- Jump Box/RDP
- SIEM
- Patch management server
- Generic Proxy Server (for business necessities, an application may need to be contacted from the IT and, because of that, could be required an intermediary)

---

<sup>4</sup>A centralized database located in the control system LAN supporting data archival and data analysis using statistical process control techniques[6]

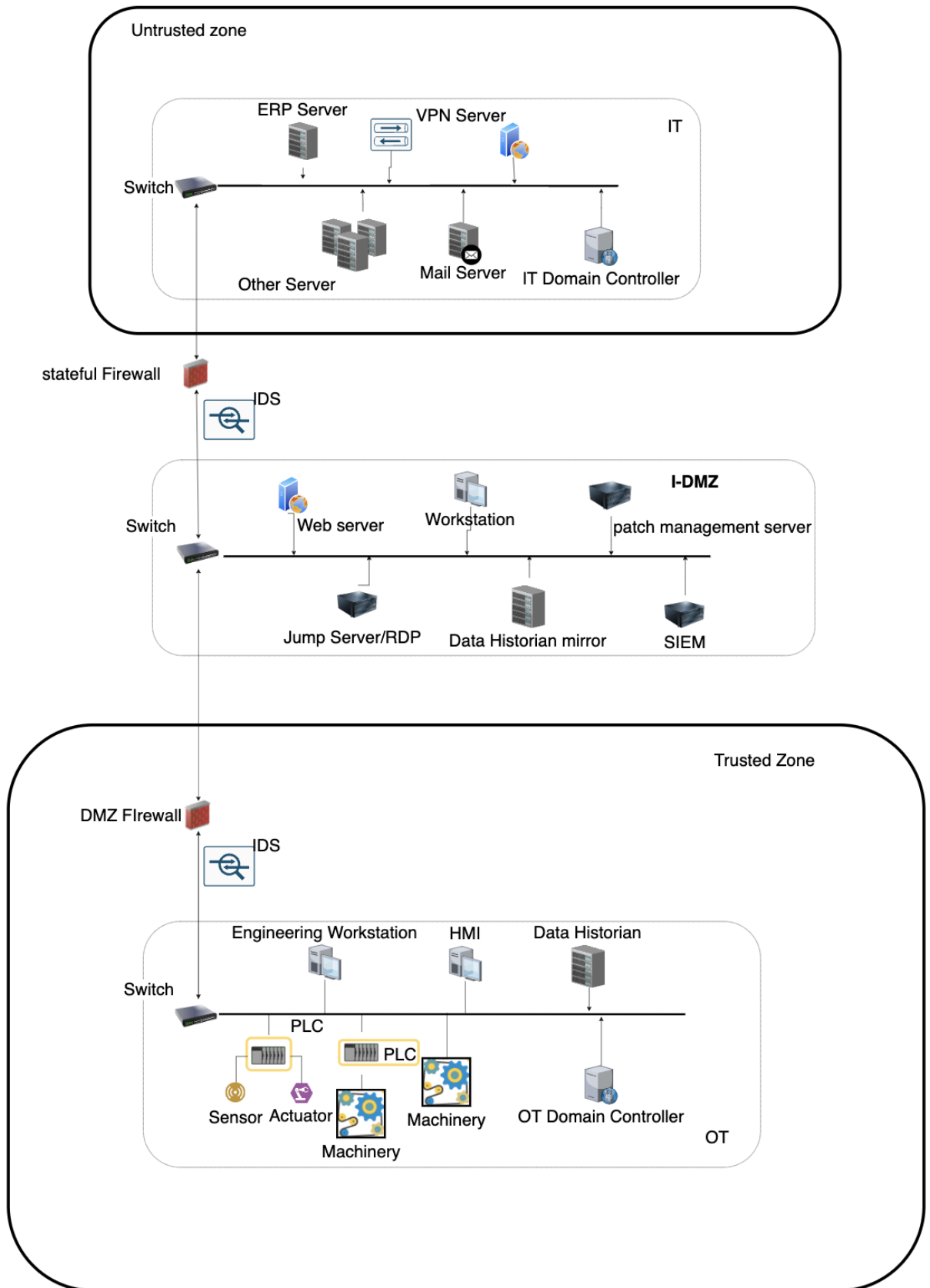


Figure 3.1: I-DMZ on prem.

### 3.1.1 Firewalls

An important role is played by firewalls that are used for the segmentation and segregation of the network and to implement strong and granular access controls. A firewall can:

- Block all communications except for those that are specifically enabled
- Enforce secure authentication of all hosts seeking to gain access to the ICS network
- Enforce destination authorization. Hosts can be restricted and allowed to reach only the nodes on the control network necessary for their job function following the least privilege principle
- Record information flow for traffic monitoring, analysis, and intrusion detection.

We have three firewalls, Figure 3.2, the first one is the first barrier from the internet directly connected to the IT network, the second one is between the IT and the I-DMZ and another one between the I-DMZ and the OT.

With this solution, we comply with the defense-in-depth principle introducing several layers between the IT and the OT, making it much more complex for an attacker to reach the corporate network.

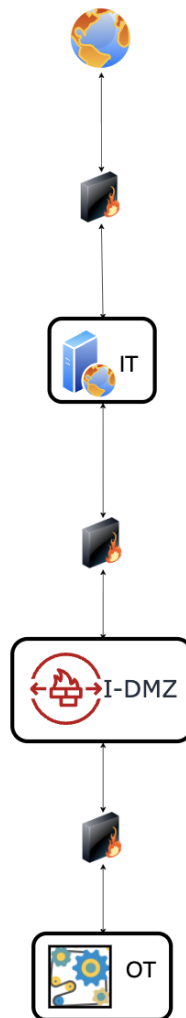


Figure 3.2: Firewalls

### 3.1.2 Remote access

Managing remote access is one of the most significant challenges in securing ICS infrastructure. Historically, as discussed, the OT network was physically separated from the IT and had no access to the internet. This is no longer true today, and remote access is required to perform several tasks.

The problem is that the old architectures have been adapted to meet the new needs without considering the risks to security. These remote connection methods often circumvent the security perimeter creating “back doors” in the Control network, this creates a huge problem, not only for the continuity of the production but also for the safety of people.

The best practice for remote access requires strong authentication and authorization policies and procedures, typically with the use of a VPN server to create an



encrypted tunnel between the remote device and the network of the company. In the proposed solution, remote access is granted through a VPN server located in the corporate network from which we can authenticate and authorize the users, any employee that wants to access the company network need a VPN client installed in his own pc (or provided by the company), once contacted the VPN server in the infrastructure he must authenticate himself with a two-factor authentication procedure. We could use the VPN server to store the user's credentials, but a more reliable and secure solution is to manage the user's credentials in the Domain controller. In this way, we can centralize the management of the authentication and authorization process and implement fine-grained policies for the access. One of the best choices is OpenVPN [59], an open-source Virtual Private Network system based on the OpenSSL encryption library and the TLS protocol. For authentication, OpenVPN provides itself a solution to manage the user credential, but for the sake of consistency, we can configure OpenVPN to interact with the Domain controller, and Windows AD DS using RADIUS (remote authentication dial-in user service) protocol [65].

Another essential point to face is the connection between areas with different privilege levels of the network. In a perfect scenario, all possible connections to the OT should be avoided, but this is no longer possible and is needed a connection. A possible solution could be to put another VPN server in the DMZ, creating a tunnel between this one and the VPN server placed in the IT. The problem with this solution is that, in this way, we create an encrypted communication between the two devices, making it impossible for the firewalls to inspect the communication packets nullifying their use. In these cases, a better solution is to exploit the DMZ and the Defense in depth principle by using a jump host placed in the DMZ. A jump host, also called Jump box or Bastion server, is a server typically placed in a secure zone that we can access from a less secure zone allowing us to add another enforcement point of authentication and authorization. When a user needs to interact with a device in the OT, he can contact the Jump server, which verifies that the employees have the right level of privilege to access.

### 3.1.3 Domain Controller

The Domain controller is used to manage all the user and network assets' credentials needed for the authentication and authorization phases; in particular, it is the server that hosts the Active Directory Domain Services and is a critical device that needs to be protected at best.

The placement of the DC in an industrial network is always a delicate decision, if a malicious actor obtains privileged access, he can modify or corrupt the AD DS database and, by extension, compromise all the systems and accounts that are managed by Active Directory. AD DS organizes network elements, such as users, computers, and other devices, into a hierarchical structure that includes the Active Directory Forest and domains in the forest.

- A forest is a security perimeter and defines the scope of authority for the administrator
- A Domain is a logical group of network objects that share the same AD Database.

In a total on-premises solution, the placement of the Domain controller is a critical choice because, as said, its compromise can become a critical problem for the security of the entire infrastructure.

Our choice is to have two different domain controllers, one for the IT and another for the OT, completely independent of each other in order to prevent that, if the IT one is compromised, this can affect the OT one giving to the attacker complete control of the infrastructure. Another important benefit of this solution is to have two different domains forcing the user to re-authenticate if they want to go from the IT to the OT, allowing them to set more fine-grained policies for access to the OT. Consider the case of an employee that, from a workstation in the IT, wants to use a service in the OT, like the interface of a PLC or the Engineering workstation, with our solution, even if the user is authorized and authenticated in the IT, he is obliged to reach the jump server and re-authenticate himself to the OT Domain controller.

Thinking of the worst case, in which an attacker could gain control of all the machines in the IT, and so also the DC, this does not compromise in any way the security of the remaining part of the network, raising the resiliency of the entire system.

As we have reported in the state of the art about I-DMZ 1.3.1, a possible addition to our solution is to have bi-directional replication between the Enterprise DC and the Industrial Zone DC. Even though this solution makes the infrastructure's management easier, it also introduces a critical vulnerability. If the attacker gains control of the IT Domain controller, the OT one can be considered compromised, the attacker can indeed create a set of credentials and copy it from the Enterprise DC to the Industrial one.

Looking to the other proposal for the management of DC, the “**Organization Domain Forest Model**” also described in the State-of-the-art1.3.1, it is an interesting solution that helps to segment at high level the network and define more fine-grained policies for each sub-domain based on the level of security that is needed. It is undoubtedly helpful for big infrastructure with many devices that require to segment the network while maintaining centralized management of all these sub-domains.

In our case, we could implement this model with two sub-domains, one for the IT and one for the OT with the possibility to define different policies for each of them, but this might not be the best solution for some reasons. The first consideration is that even if the Domain Forest Model allows to segment of the infrastructure, creating perimeters on the basis of the resources and the authorization level required, we always have the Domain owner that has full access to all the resources in all the sub-domains, this means that if an attacker gets hold of the credentials of the Domain owner he has essentially full control over the entire network, introducing a critical vulnerability for the security of the infrastructure. Another consideration about this model is that it introduces another level of complexity for the management of the network that could lead to introducing vulnerabilities, the KISS (keep it simple and stupid) principles should be followed whenever possible.

So, finally, after these considerations, in our case, the best solution is to have two separate Domain Controller, one independent from the other with two different forest root domains.

### 3.1.4 Data Historian mirror

A Data Historian server is another crucial component in an ICS Infrastructure, it stores and logs all the data that SCADA, PLC, or any other OT system aggregates, making it possible for the employees to analyze and use the information about the production.

It can also automatically generate periodic reports that can be used to understand better what happens in the plant and make decisions accordingly, collect critical time-series data for various calculations, estimations, and statistical processes producing information to benefit a multitude of enterprise-wide processes and applications. Because of the nature of the information stored in the Historian, it could also store industrial secrets. Because of that, it could become a target of data exfiltration attacks for competition or by generic malicious actors, so it

is crucial to protect this device well. Additionally, if an attacker manages to compromise the Data Historian, it can bring to a loss of visibility of industrial processes causing damages to the industrial line.

Because of those considerations, in our solution, we have just one primary Historian server placed in the OT network and a Historian mirror in the I-DMZ that pulls the data from the OT one. In this way, we have another level of protection provided by the firewall placed between the IT and the I-DMZ.

Another benefit of this solution is that we have a “point of disconnection” for the attacker. If we understand that a malicious actor has succeeded in exploiting a vulnerability and compromising an IT machine, we can block all the communications from the IT, disconnecting it from the rest of the network. So, with our solution, the I-DMZ and the OT can still work without too much trouble, while the attacker has been blocked in the IT.

This solution is a variation of the best practice proposed by CISCO and Rockwell Automation reported in the State-of-the-art 1.3.1 This solution is certainly efficient from the availability point of view, and makes it easier for IT users to access the data of the Historian but, at the same time, makes it also easier for an attacker that manages to penetrate the IT, to access the data of the Historian and exfiltrate sensitive data.

So the problem, in this case, is that the Historian is placed in the weakest part of the network, the IT level is the most vulnerable and the typical starting point of the majority of the cyberattacks to an Industrial Architecture.

### 3.1.5 Patch management server

The Patch Management server is a machine dedicated to the management of updates, in particular for the OT services.

As stated in the article [36], software update systems must guarantee high availability, integrity and security, even in presence of cyber attacks. Keeping the devices updated is one of the essential points for security but can become a weakness if the update process is not managed correctly. Allow devices to be directly connected to the vendor site, and download the updates directly, is not the best solution, and a man-in-the-middle attack can be easily used to install malicious software. A technique that can be used to secure the update process is the so-called “sheep dip” [73].

The sheep-dipping operation is generically used when a USB drive or a hard drive needs to be plugged into a device inside the network; these are first plugged into a specific device with two folders and a tool that can detect malicious software. The first folder is a read-only folder that extracts the files from the drive analyzed by the security tool; when the files pass the analysis, these are copied into the write-only folder and in a new drive.

This simple procedure could have prevented the famous Stuxnet attacks that, as reported, traveled on USB sticks.

Stuxnet is not the only such example, in 2008 a USB flash drive containing malicious code was plugged into a laptop connected to United States Central Command, and it took the U.S. Department of Defense over a year to remove the resulting outbreak [33].

In 2012 two U.S. power plants were infected by malware that was introduced to the network accidentally via a USB drive. The BBC reported that "ICS-CERT said it expected a rise in the number of similar attacks".

In 2016 Bloomberg reported [32] that hackers were increasingly targeting critical infrastructure facilities using infected USB drives.

In our case, we can use the sheep dip procedure to sanitize firmware and software updates before installing them, we can download the updates in the patch management server, and after being insured that they are safe, we can install them in the right device.

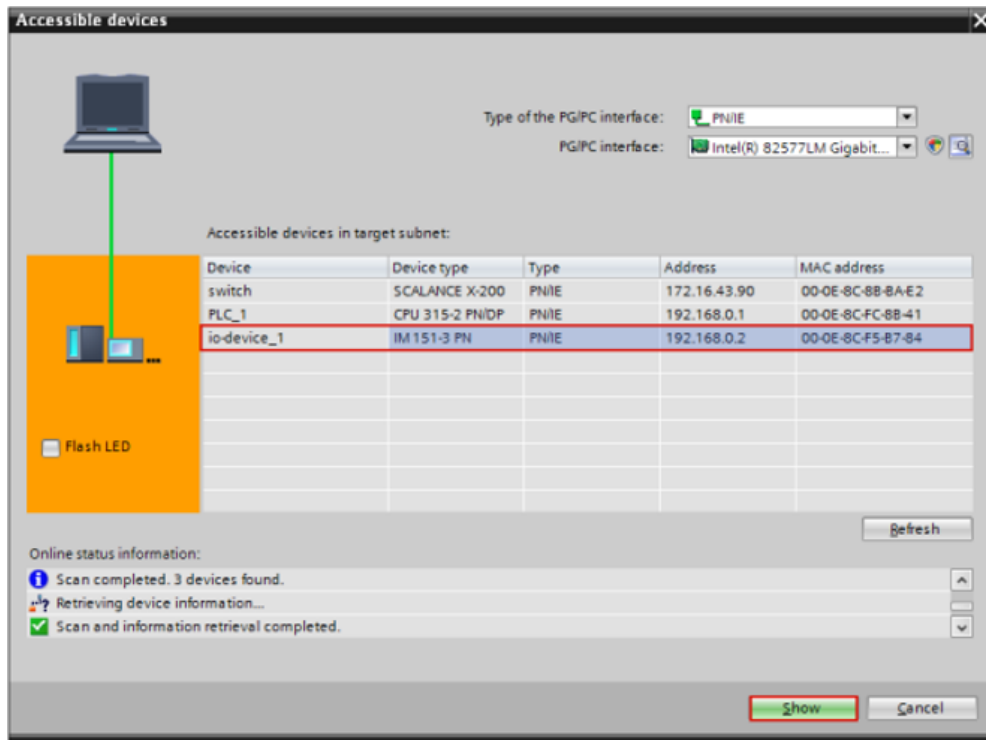
In this way, we avoid direct connection from the PLCs or any other OT devices to the vendor server. This solution is not particularly complex to implement, the idea is to have a machine from which, when there is a new software/firmware update, we can download and install it.

In an Industrial environment, we can have devices from several vendors, this means that the patch management server must have installed all the necessary software to update the devices (TIA Portal, Windows update, ...). An example could be the updating process of SIEMENS PLC using the TIA Portal .

TIA Portal (Totally Integrated Automation Portal) [68] is the proprietary engineering framework by SIEMENS for the configuration and management of their products.

We need the TIA portal installed in the patch management server from which we can inspect the network to identify the devices and select the ones to update.

At this point, we can download the correct software/firmware update from the SIEMENS archive. Once the software is downloaded, we can inspect the files with



**Figure 3.3:** TIA Portal, accessible devices.

a static analysis tool to ensure that they are safe, and then, we can install it in the PLC. Considering that the typical system in an ICS environment is based on Windows, a clear choice is to use a software like Microsoft Defender for IoT [46], a tool that works with several IoT, OT and ICS devices to secure them.

It is also interconnected with other Microsoft security software and external SOC to find also the most recent vulnerabilities. Microsoft Defender for IoT is also based on Binwalk[66], a software that can be considered a standard for analyzing and reverse engineering firmware. Once confirmed that the update is secure, we can start the installation process in the PLC.

### 3.1.6 Digital Twin

Another possible addition is the use of Digital Twin (out of the scope of this thesis). A Digital twin is a virtual model that reflects a physical device, environment, or everything that needs to be tested. The use of this solution has spread a lot with the growth of the industry 4.0 and. In our case, we could create a digital twin of our OT and use it to test updates before installing them in the real devices, this could help a lot because the operations of the OT typically depend on the entire production.

Independently from malicious actors, it is not rare that an update process of a

PLC goes wrong, causing the stop of the production and the need for manual intervention, and a significant loss of money. This solution began to spread with industry 4.0 and IoT thanks to its flexibility and the capacity to test in a secure environment, a solution that will then be deployed in the real infrastructure. This is fundamental in the industrial environment, where a stop of a couple of minutes could cause a significant loss of money. This is why administrators are typically reluctant to patch, update or modify the infrastructure.

Digital twins can also be used to conduct extensive cybersecurity assessments to test the reliability of the system and the ability to withstand a cyber-attack. They can play an essential role during the phases following the attack for forensic analysis, data captured during this phase can be used to acquire new information about attacks and improve defensive models. As shown in this article [13], the use of a Digital Twin for continuous verification in software security has led to a reduction of about 50-80% of the vulnerabilities.

## 3.2 Zero Trust Architecture

The proposal described in the previous paragraph leverages all the solutions and best practices for the cybersecurity of the industrial sector based on perimeter. However, even if it can protect the architecture from attacks well, this is made possible at the expense of the connectivity and implementation of new technologies. If we start to add to the architecture all the functionalities that the companies need to be up with the times, just think of the exportation of the historian data to the cloud for advanced processing, the I-DMZ security model will be broken, and it is no longer respected the Defense-in-depth principle, with all the ensuing consequences.

This chapter will describe our innovative proposal of Zero Trust Architecture for the industrial sector. It overcomes the security issues of the traditional I-DMZ solution, allowing, at the same time, to implement all the new services and functionalities that the industrial sector needs without compromising its security and safety. It is based on a new approach for this sector that avoids any concept of perimeter thanks to a strong authentication and authorization mechanism, allowing it to monitor and control any communication inside the network.

In a Zero Trust architecture, one of the assumptions at the basis is that the entire enterprise network is compromised. Therefore, the notion of “Trust” does not exist anymore, and consequently, the concept of perimeter loses its meaning [4]. Looking at the variations of ZTA approaches proposed by the NIST 1.4.1, our

solution, showed in Figure 3.4, can be considered as a Hybrid between the "ZTA Using Enhanced Identity Governance" and the "ZTA Using Micro-segmentation", we have strong access management and user/device identity governance, but we also need to segment the network to protect the critical devices of the OT from any potential threats.

Although the classical Micro-segmentation implementation is very used, it requires the concept of perimeter, even if small, which implies consequently the concept of trust inside this perimeter going against the basic concepts of the Zero Trust Approach; this is the reason why our network segmentation is per-device.

The implementation is based on two fundamental components, the Access Proxy (AP) and Next Generation Firewall (NGFW). They permit the implementation of fine-grained Authentication and Authorization processes and segment the network, to be sure that all the communication, both from outside and from inside, can be monitored and controlled. It also provides continuous authentication.

The network can be divided into three main areas: Access Control, IT, and OT. The different areas are connected by the NGFW, which plays the role of Policy Enforcement Point and Policy Administrator to monitor and control any communication in the network.

Thanks to the Zero Trust approach and the fine-grained segmentation, our solution minimizes the possibility of lateral movement. If for the IT sector, those types of solutions start to be spread, in the industrial sector are pretty new, and there is a growing need for new ideas and solutions that meet the specific requirements of ICS.



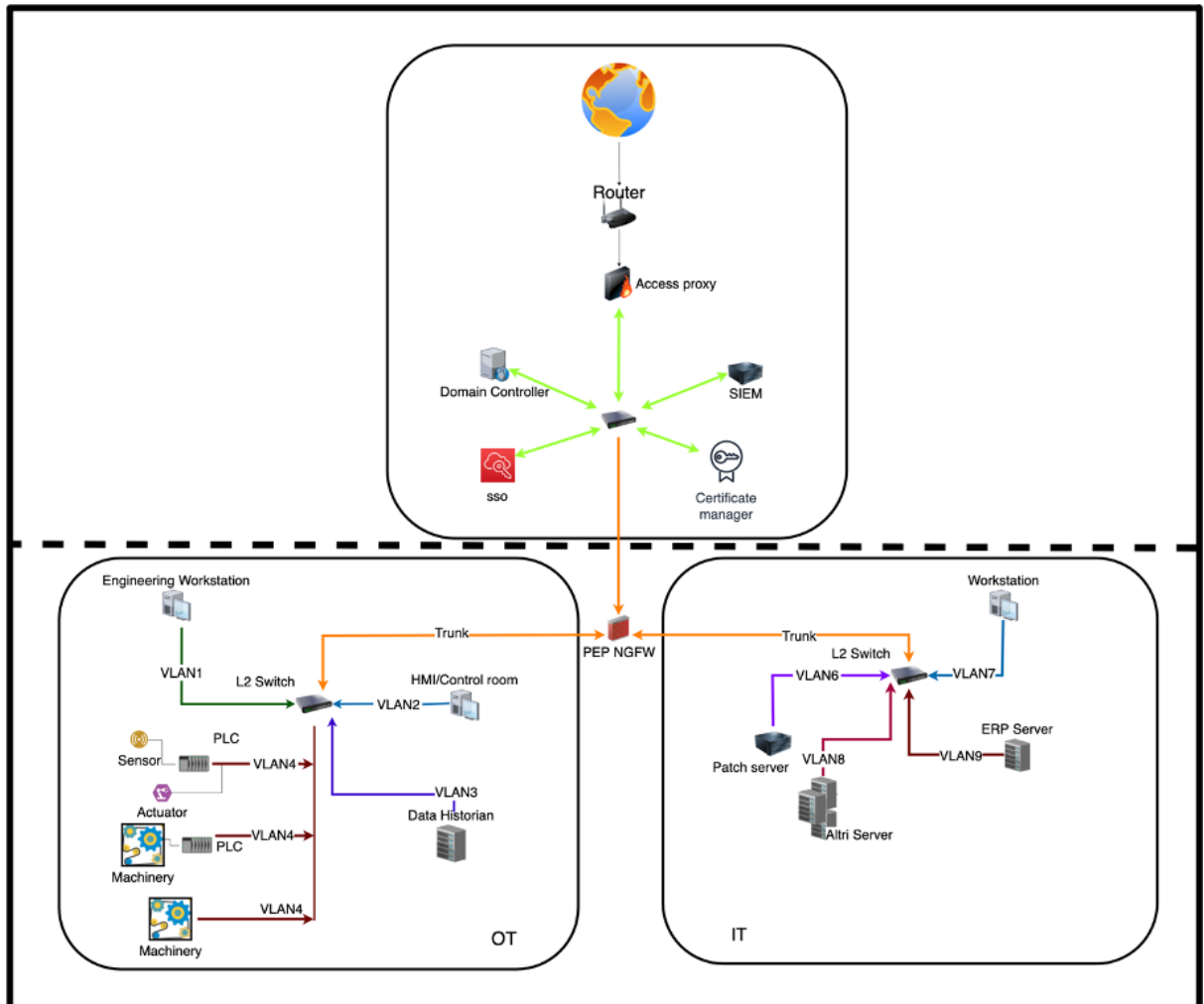


Figure 3.4: Zero Trust on-premises architecture.

### 3.2.1 Access control part

The first part is the one that manages the access to the network; the core is the Access proxy that provides authorization and authentication per access.

To provide this functionality, it is placed in the same VLAN of:

- Certificate manager
  - For devices authentication
- SSO (Single Sign-on) server
  - The Single-sign-on server provides user authentication and authorization
  - It communicates with the domain server

- Domain Server
  - Store all the credentials and information about each user
  - Used both from the SSO and the NGFW
- SIEM (Security Information Event Management)
  - It memorizes everything that happens in the network
  - Provide real-time analysis and alert

The combination of these services is used to grant secure access to the network.

### Access Proxy

An essential point for the security of any infrastructure, but especially for a Critical one, is how to manage the remote accesses of employees [19]. The typical solution is to use a VPN to create an encrypted tunnel between the remote device and the company's network, but, looking at the State-of-the-art, this solution will phase out in favor of solutions based on an Access Proxy.

The AP has the role of Policy Engine. It provides authorization and authentication for users and devices based on fine-grained dynamic policies that take into account the user and devices information (the type of device, if it is patched, etc.), but also the context like the moment of the day or the general risk level of the infrastructure.

A problem with the traditional VPN approach is that it is a static approach, we decide to give or not access to the network, and we cannot manage per-service requests. Another drawback of a VPN solution is that it introduces another point of vulnerability in the network that requires configuration and maintenance.

As highlighted in the report [76], especially during the last few years, with the pandemic that led to massive use of VPNs for remote work, these are becoming one of the main targets of cybercriminals with new vulnerabilities that are coming up every day, **“Corporate VPN is an aging technology as organizations shift to more cloud-based services... However, in the wake of the global coronavirus pandemic, companies realize that they have to fundamentally change the way they work.”** - (Rob Smith, Senior Director Analyst, Gartner). Threat actors are targeting VPNs, as made evident by the countless new articles about VPN exploits and almost 500 known VPN vulnerabilities listed on the CVE database. With the Access proxy solution, there is no direct routability between users and applications, instead, all accesses are routed through the AP,

which can keep dynamic decisions. The solution draws inspiration from Google's BeyondCorp implementation and is based on a front-end Access proxy to which are directed all the external requests. In front of the AP is placed a router that is used to control the number of requests and avoid DoS attacks.

It is clear that this is an ideal implementation, the Access Proxy plays a crucial role in the entire infrastructure and can be considered a Single point of failure. Because of that, in a real scenario, we could replicate it by increasing the number of Access Proxies and placing a load balancer in front of them.

The real core of the AP is the Policies engine, a fundamental process of a ZTA that executes the Trust algorithm and is responsible for the ultimate decision to grant access to a resource or not.

The AP is connected to the SSO system, which has the task of authenticating the user through using the information stored in the domain controller, and to the certificate manager that is responsible for the authentication and authorization of devices.

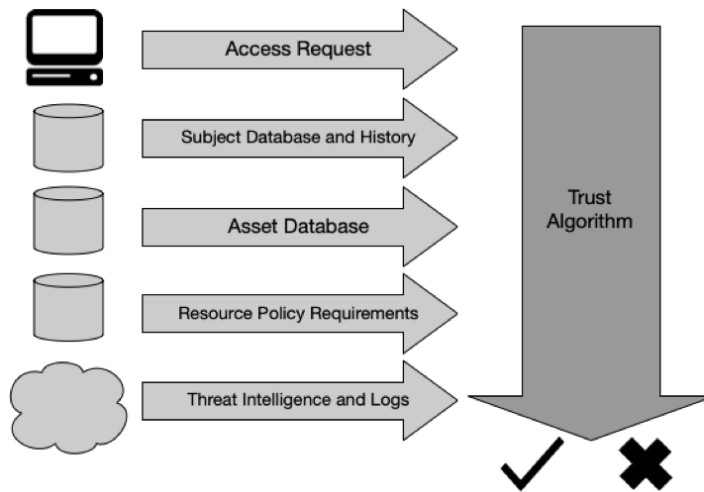
### **Policies Engine**

As previously said, the policies engine is the core of the access management. It aggregates all the information about the user, device, and context to authenticate him and dynamically decide the associated authorization level. It is also the component that executes the Trust Algorithm process, through which it is decided to grant or not access to a specific resource every time a request arrives. This algorithm comprises several steps, which must be checked to approve a request. Looking at the NIST definitions of Trust algorithms 1.4.1, our solution falls into the category of a criteria-based solution with contextual information. Each user has the right to access specific resources, and his behavior is continuously monitored during communications.

The TA consists of the following steps:

1. The user sends a request for a resource that is directed to the Access Proxy.
2. The device provides the certificate to the Access Proxy
3. The AP receives the request and redirects the user to the ID management system (SSO).
4. The user provides the primary and secondary authentication information.
5. The ID management system asks the Domain controller and validates the user credential.

6. If the user is authenticated, the ID management system release a token with the user information and sends it back to the AP.
7. At this point, the AP made the authorization check
  - Identify the user’s authorization level based on the identity and other contextual information.
  - Identify the device and decide if the request can be satisfied
8. If all these checks are passed, the AP allows access to the right resource.



**Figure 3.5:** Trust Algorithm NIST SP 800-207.

As said, the decision to grant or not access to a resource is based on several contextual information. An important role is played by the Threat intelligence systems <sup>5</sup> (Embedded in the NGFW) that can help to identify, for example, if a device is affected by some vulnerability or is not patched, dynamically changing the “trust” level of a specific device.

Other information could be the number of accesses of an employee to a specific resource, the hour of the access, and the location of the IP from which the request come from. All those information are stored to create a “context” and analyze the behavior of employees, sending alerts if something unusual happens.

<sup>5</sup>A Threat Intelligence System/Service is a technology solution that collects, aggregates, and organizes threat intelligence data from multiple sources and formats. It provides to security teams information about known malware and other threats, powering efficient and accurate threat identification, investigation, and response

## Certificate manager (PKI)

The main components of a PKI<sup>6</sup> are:

1. **Certificate authority (CA):** Issues entity's certificate and acts as a trusted component within a private PKI. Any certificate issued by the CA is trusted by all entities that trust the CA.
2. **Certificate:** A digital document, signed by a CA, and used to prove the owner of a public key, within a PKI. The certificate has several attributes, such as usage of the key, Client authentication, Server authentication or Digital signature, and public key. It also contains the subject name, which is information identifying the owner, such as a DNS name or IP address.
3. **Registration Authority (RA):** Receives certificate signing requests and verifies the identity of an end entity. The RA will approve a request before that the certificate can be issued by the CA. This is a critical stage of the process, and it often involves a procedure to enroll end entities into the PKI.
4. **Validation authority (VA):** A VA allows an entity to check that a certificate has not been revoked. The VA role is often carried out by an online facility hosted by an organization that operates the PKI.
5. **Secure storage:** A method to securely store a private key is required for both the Certificate Authority (CA) and end entity to protect the key from compromise.
6. **Public/Private key pair:** A private key and associated public key are mathematically coupled together. The public key can be shared widely. The private key proves ownership of the identity and must be kept secret.

The RA is not represented because the release of a certificate is done when the device is assigned to an employee. The certificate manager plays the role of the VA, that host also a Device Database.

When a user sends a request for a resource and connects to the AP, we need to identify the device from which this request arrives. To do that, we need that all the managed devices provide their certificate, released by the company's Certificate Authority.

---

<sup>6</sup>A public key infrastructure (PKI) is a set of roles, policies, hardware, software and procedures needed to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption

The easiest way to implement a CA is to use OpenSSL, an open-source toolkit for general-purpose cryptography and secure communications that can be considered a standard for these tasks. Another important characteristic of OpenSSL is that it is compatible with all platforms and can be integrated with Azure, AWS, or other PaaS. We can use X.509 certificate to identify devices, a digital document that contains the device's public key and can be used to verify that a device is what it claims to be.

Two are the most important resources to manage our CA:

- Private key
  - Used to sign our root certificate
- Root certificate
  - The most important certificate, is used to sign device certificates and is important to store it in a secure place.

Create and manage a PKI with OpenSSL is relatively easy and can be done in a few steps:

1. Generate our private key used to self-sign the root certificate
2. Generate and sign our root certificate with the private key
3. Generate and install the device certificates and the relative private/public key for each machine

### **SSO (ID management)**

An SSO (Single sign-on) is a mechanism that allows users to be authorized and authenticate to different services or applications with the same credentials[64]. Because identifying a user is a pillar of our implementation and of security in general, we decide to use a Single sign-on system that allows the user to use the same credentials to access all the company's resources. It is up to the policy engine to determine the role of the specific user and if he has or not the right to access the resource.

Having a centralized authentication portal is helpful to create a robust access system, many times, companies use several different authentication mechanisms with different credentials. This, despite appearances, does not increase the level of security; instead, it could become a security issue because it can become tedious

for the user, leading to the use of weaker credentials. As with any technology, SSO systems are also affected by vulnerabilities, mainly due to centralized design. Nevertheless, as described in this article [35], if well implemented, it can ensure an optimal level of security and flexibility.

With this solution, we increase not only the security of our system but also the user's experience. Users are automatically connected to the correct service and do not have to enter their passwords repeatedly. As previously said, the SSO system is directly connected with the Domain controller that stores all the information about the users, like credentials, the role inside the company, and all the information that could be used from the policy engine.

The authentication is at least a 2FA to ensure the proper level of security. When the user has provided his 2FA credential, the SSO server contacts the Domain controller to validate these credentials. If the user is recognized, the SSO generates a token with all the valuable information for the PE to determine the user's access level. Following the Zero Trust principles, the token must be signed, even if the SSO is directly connected to the AP. Because of that, also the SSO system requires a certificate released by the CA. This token is passed to the AP to continue the execution of the Trust algorithm.

### **Domain Controller**

As in the case of I-DMZ, the Domain Controller plays a crucial role in the security of the entire infrastructure. The implementation is similar to the I-DMZ one. However, in a ZTA, this role is much more critical because the authentication and authorization of employees are the core of the architecture. In this case, the user's authorization is for a specific service and not for a subnet or a group of resources. This means that any time that someone needs to interact with a service or a device, the Domain controller must be contacted to be sure that that specific user has the proper privilege for that resource. Because of that, this device stores all the information about an employee, and it is essential to keep it updated whenever an employee changes role or leaves the company.

In our implementation, the Domain controller is directly connected with the ID management system (SSO) for the first authentication and authorization phase and with the NGFW that needs to determine the authorization level of a user at any request and to authenticate continuously him.

It is placed in the same VLAN of the SSO and the Certificate manager.

### 3.2.2 SIEM

The SIEM is another powerful device for the security of a network. It aggregates information from the entire network and brings it together to have a centralized vision.

It provides visibility of the entire network, allowing security teams to gain attacker insights derived from well know attacker tactics, techniques, and procedures (TTP) and Indicators of Compromise (IOCs).

It is a really useful device to get an overview of the entire network and, overall, to facilitate the forensics operation. It is not specific of a Zero Trust approach (this is because it is also present in the I-DMZ implementation), but fits perfectly with the principles of continuous monitoring and network visibility typical of the Zero Trust. It also provides other functionalities that overlap or integrate with NGFW functionalities like:

- Threat intelligence and detection
  - It uses internal or external threat intelligence sources to detect threats in emails, cloud resources, application
- UEBA (User and Identity behavior analytics)
  - Analyze the user behavior to determine a “normal” one.
  - It can raise the alarm in case of suspicious actions
- Data aggregation
  - gathering vast amounts of data from various applications and databases in one place.
- Data normalization
  - SIEM allows all the disparate data to be compared, correlated, and analyzed.
- Data analysis/security event correlation
  - Determining potential signs of a data breach, threat, attack, or vulnerability.
- Supports compliance and logging.



Just like the NGFW can be used to determine the system's state and dynamically modify the security posture of the systems according to predefined rules. In our infrastructure, the SIEM is directly connected with NGFW, which can be considered as the crossroad of the entire infrastructure; in this way, the connections can be easily monitored.

### 3.2.3 Segmentation

Segmentation plays a fundamental role in the security of the infrastructure, allowing it to define policies to control the interaction with all the devices.

Without network segmentation, if an attacker manages to penetrate a subnet of the infrastructure, we can consider all the machines inside that infrastructure compromised. It helps in deterring lateral movement between different machines and became fundamental to grant per-device access.

In our implementation, we have different VLANs, one for each device, both IT and OT and, as in the I-DMZ, we have a stronger separation between the IT and OT, among which is placed the NGFW. This separation technically would not be necessary because, with a VLAN, a network is segmented at layer 2, but like any technology, also VLAN can be affected by vulnerabilities like the so-called VLAN Hopping [3] that allows an attacker to bypass any layer 2 restrictions.

With this physical separation, we introduce another layer of security between IT and OT. This means that an attacker, even if he manages to compromise an IT device, should pass through the NGFW to access the OT devices. With this solution, the communication between devices in different VLANs must be managed.

In general, there are three different ways to allow communications between devices in different VLANs:

- Configure a router and connect a single interface to a switch per VLAN configured.
- Configure a router to use IEEE 802.1Q and connect to a switch via a trunk.
- Configure a Layer 3 switch.

The typical solution is to use a layer 3 switch capable of managing IP routing. However, because we want to force all communication to pass through the NGFW, we use another procedure called VLAN Insertion based on trunk connections which also allows using a layer 2 switch that is much cheaper and typically faster than a Layer 3 one.

### 3.2.4 Next Generation Firewall

The other core component of our implementation is the New Generation Firewall, which is a security device that provides, in addition to the normal functionalities of a stateful inspection firewall like:

- Keep track and monitor the state of a connection.
- Block traffic based on state, port, and protocol, and filter traffic based on administrator-defined rules.
- Stateful packet inspection
  - Examine the content of the packets to ensure that the data are coherent with the scope of the communication and the other data exchanged in the communication
- Monitor TCP connections

a lot of other security functionalities as:

- Integrated intrusion prevention
- Application awareness and control to see and block risky apps
- Threat intelligence sources
- Upgrade paths to include future information feeds
- Techniques to address evolving security threats

It can block advanced malwares and integrate several functionalities like Intrusion Prevention/Detection, Threat intelligence across users, hosts, networks, and devices in a single device. It can dynamically adapt to the system's state and react fast if something goes wrong to improve the security posture and reduce the incident response time.

A NGFW is a powerful device, a great ally for the security of an infrastructure that supports a holistic approach to security, also interacting with the other security devices. At the same time, it requires significant efforts by the administrator to define all the policies carefully to put them into practice.

This holistic approach fits with a ZTA solution that requires not only new technologies but a totally new approach that must take care of any aspect of the system. In our implementation, it is directly connected with the Domain controller

to authenticate and authorize the users continuously.

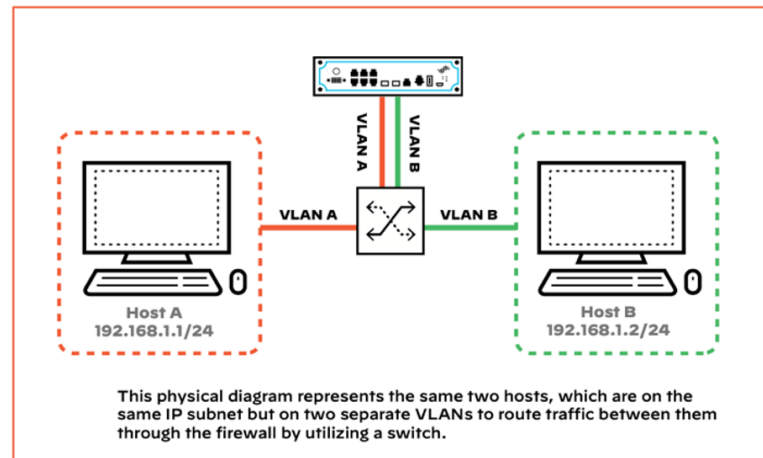
After the first authentication phase provided by the Access proxy, any time a user wants to interact with another resource in the network, the request is directed to the NGFW, which inspects the packets and determines if that specific user can or not interact with that specific device. From the implementation point of view, we want that all the communication passes through the NGFW, not only the ones between the IT and the OT but also the communications between all the different devices, to implement fine-grained access control previously described. To do that, the NGFW is placed physically between the IT and the OT to force the path of communications, while to route the intra-device communications, we use a technique previously mentioned called VLAN insertion.

### 3.2.5 VLAN Insertion

As described in the Palo Alto document[60], VLAN Insertion is the logical placement of one device between two others without the need for physical recabling of the original devices or the introduction of additional switches.

The use of VLAN insertion provides a method to introduce or remove a device from the data path with minimal disruption, in this way, we can force communications between the devices to pass through the NGFW. This solution leverages the IEEE 802.1q Trunking protocol that allows the creation of a Layer 2 link between switches or other devices like NGFW and is used to allow communication between different VLANs.[26]

From the technical point of view, when a trunk line is installed, in our case between the switches and the NGFW, the Ethernet frame is modified by “tagging” it directly through the trunk line. This solution allows us to segment the network without recabling or reassigning IP addresses. In this way, we can create different VLANs in the same subnet.



**Figure 3.6:** VLAN insertion Paloalto networks Applying VLAN Insertion in ICS/S-CADA.

Essentially the NGFW act as a Virtual conduit between the different devices monitoring everything that is sent or received. It can allow or deny traffic based on the zone's VLAN tag to enforce security or dynamically restrict access to a VLAN if it is needed.

### 3.2.6 Variation for performance improvement

A possible problem with this implementation of the NGFW could be the performances. In an Industrial System, it is crucial that all the communication, especially between OT devices, are not subject to excessive delays.

This is a typical challenge for network and security engineers to balance the security and performance requirements.

In our case, all the communications between two devices must pass through the NGFW, which might seem a bottleneck for the performances. Suppose we use the Palo Alto Networks NGFW PA-5200 Series as a reference, for example, the PA-5250. In that case, as we can read from the relative datasheet [61], the device supports a throughput of 32.8Gbps, a quite sufficient capacity for the majority of networks.

Table 1: PA-5200 Series Performance and Capacities				
	PA-5280	PA-5260	PA-5250	PA-5220
Firewall throughput (HTTP/appmix)*	49.5/55.2 Gbps	49.5/55.2 Gbps	32.8/36.7 Gbps	13.9/15.6 Gbps
Threat Prevention throughput (HTTP/appmix)†	25.4/31.4 Gbps	25.4/31.4 Gbps	16.9/21.4 Gbps	7.1/8.8 Gbps
IPsec VPN throughput‡	26.3 Gbps	26.3 Gbps	18.4 Gbps	9.5 Gbps
Max sessions	64M	32M	8M	4M
New sessions per second§	505,000	505,000	368,000	155,000
Virtual systems (base/max)	25/225	25/225	25/125	10/20

Note: Results were measured on PAN-OS 10.2.  
 \* Firewall throughput is measured with App-ID and logging enabled, utilizing 64 KB HTTP/appmix transactions.  
 † Threat Prevention throughput is measured with App-ID, IPS, antivirus, antispysware, WildFire, file blocking, and logging enabled, utilizing 64 KB HTTP/appmix transactions.  
 ‡ IPsec VPN throughput is measured with 64 KB HTTP transactions and logging enabled.  
 § New sessions per second is measured with application -override, utilizing 1 byte HTTP transactions.  
 || Adding virtual systems over base quantity requires a separately purchased license.

Figure 3.7: PA-5200 Series Performance and Capacities Paloalto Networks NGFW.

Despite this, today, exist many types of ICS environments with different needs in terms of communication performances.

There are cases in which the requirement in terms of communication delay between OT devices is under 1µs; in these cases, our implementation could not support the requirements because the NGFW introduces an unavoidable delay due to the processing time.

A possible solution to this problem is to use an Industrial Security Appliance like the ‘SCALANCE SC-600’ from SIEMENS.

This device offers the functionality of switching but also of Firewall, and it can provide ‘Stateful inspection’ functionality and support up to 1000 firewall rules.

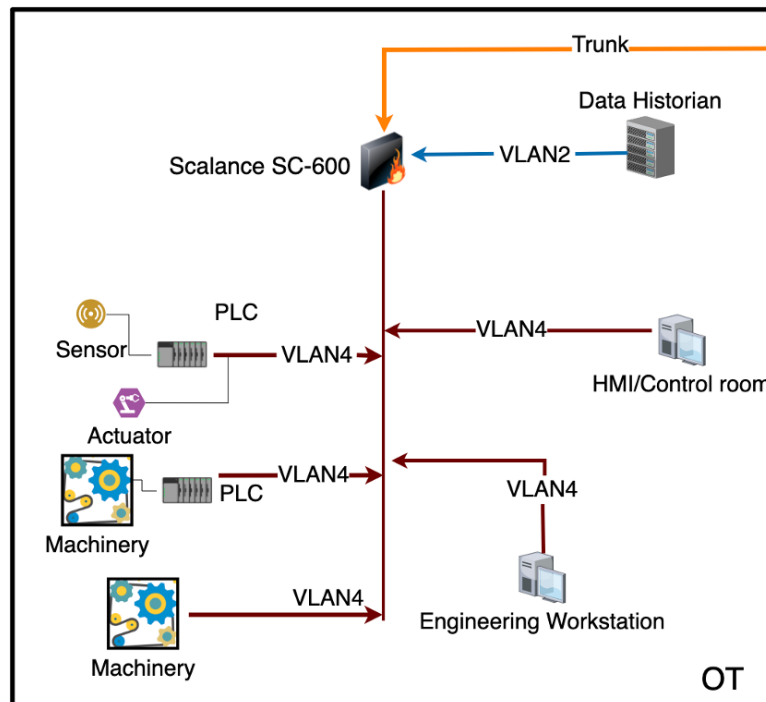


Figure 3.8: OT alternative.

As we can see from the Figure 3.8, this alternative solution requires that the Engineering Workstation and the HMI/Control room are placed in the same VLAN of the PLC to avoid any delay due to the communications.

It is clear that this solution decreases the level of security of the network because the intra-OT communication will no longer be inspected by the NGFW. In this solution, the SCALANCE sc-600 plays the role of another PEP introducing another level of security inspection. In case of problems, it can also block all communications to and from the OT, providing isolation from external connections. Also in this case, the Data Historian remains in another VLAN, and the NGFW inspects all the communications to and from it, this is because the throughput of the NGFW can widely support the requirements of the Data Historian.

The choice of this alternative solution should be carefully assessed based on an analysis of the performance requirements of the specific implementation.

### 3.3 Cloud solution

This paragraph can be considered as an extension of the previous one. However, even more companies use Cloud services, and a real innovative architecture can not be imagined totally on-premises. Therefore, we propose a cloud-based Zero Trust implementation where the deployment and management parts are made even more efficient and easier to implement using Paas Cloud solution. The transition to a Zero Trust solution does not have to be necessary a problem or too challenging, and Cloud can be an important ally in this process.

Today cloud solutions are quite normal for IT infrastructure and services; most organizations are moving from an on-premises/hybrid solution to a total integration with a cloud platform. With the advent of the Smart grid, Industry 4.0, and IoT, also for OT networks, the number of cloud solutions has increased, and more and more companies are considering the transition to the Cloud. However, this transition is challenging, and if it is not carried out systematically, it can bring problematic and dangerous situations.

At the same time, this digital transformation can bring many benefits. From the business perspective, IIoT, new technologies, and Cloud can increase the productivity of an industrial system and enable rapid response to competitive threats and easier adoption of new ideas. From the security point of view, this transition could be seen as a problem or an opportunity to improve resiliency, and help companies address existing issues with cybersecurity strategies that continue to leave critical assets at risk of a serious cyber incident.

Another important trend that became crucial in industry 4.0 is fog/edge computing. Edge computing is a new paradigm that extends the Cloud platform model by providing computing resources on the edges of the network [31]. Data are aggregated, analyzed, stored, and processed between the source and the cloud infrastructure, reducing the data transmission overheads and subsequently improving the computing performances in Cloud platforms by reducing the requirement to process and store large volumes of superfluous data.

Because of that, the so-called Edge gateways, the devices that execute these processes, became critical for the security of the entire network, being “doors” between the internal and external worlds. Another important choice to take for this digital transition is the deployment model of Cloud that we want to use.

There are different models for cloud solutions:

- Public Cloud
  - The cloud infrastructure is provisioned for open use by the general public (mega-scale infrastructure). The advantage is that there are many different providers so that we can find an offer suitable to our needs.
- Private Cloud
  - The cloud infrastructure is provisioned for exclusive use by a single organization (enterprise owned or leased). No one can access it but the internal people.
- Hybrid Cloud
  - In this case, the cloud infrastructure is a composition of two or more different clouds that remain separate entities but are bounded together by standardized or proprietary technologies that enable data and application portability.
- Community cloud
  - Indicate a shared infrastructure for a specific community like institutions can try to evaluate more the local communities: government clouds could be more careful to privacy issues, for example.

The choice between these different deployment models should be made according to the network’s requirement, the company’s business goals, and, not less important,

the budget allocated. Each of these solutions has its pros and cons. A Public cloud solution offers less than a private solution in terms of performance because resources are shared between different tenants. In contrast, in a private solution, these are reserved for a single client. On the other side, we have that a public cloud offers more, in terms of reliability and disaster recovery, because the infrastructure is unavoidably bigger and more geographically distributed.

The Hybrid solution is used by companies that want to keep some service on-premises or on different clouds for different reasons like costs, distrust in Cloud, compliance requirements, or infrastructure needs.

As we can read from the Paloalto Networks report [62], more than 73% of companies now have applications or infrastructure in the Cloud. Most of them say that the main problem with cloud environments is to secure the accesses and loss of insight into:

- Who is accessing their applications and data, or even what devices are being used to access them (e.g., smartphones, tablets, laptops, etc.), since most of their assets are on third-party infrastructure.
- How data is being used and shared.

To face these problems, many companies use several different solutions, as shown in Figure 3.9, creating a fragmented cybersecurity architecture.

Location	Technology Used for Access
On-premises data centers	Remote access VPN
Private applications (data center, hybrid cloud)	Software-defined perimeter
Public cloud	Inbound proxy or virtualized firewall
SaaS applications	CASB proxy

**Figure 3.9:** Different access technologies.

So it is important to make a choice, clear and simple, which allows to define an effective cybersecurity strategy

### 3.3.1 Network description

Our solution, described in Figure 3.10, defines a strategy applicable to different situations to show how the Cloud can help in the transition from a "traditional" perimeter-based solution to a Zero Trust one.



A clear choice has been to bring into the Cloud the ERP server, the Data Historian, the Domain controller (that is replaced by Azure Active Domain), and the software of the Engineering workstation. For several reasons, other components, like the HMI, cannot be moved to the Cloud.

In particular, for these types of components, it is required to stay on-site because they need to be directly connected with the industrial plant. They typically need to perform operations on plants in real time that cannot tolerate delays that the Cloud could introduce; it is necessary that they are always accessible by the company operators in case of emergency. However, it is possible to implement remote access for routine operations.

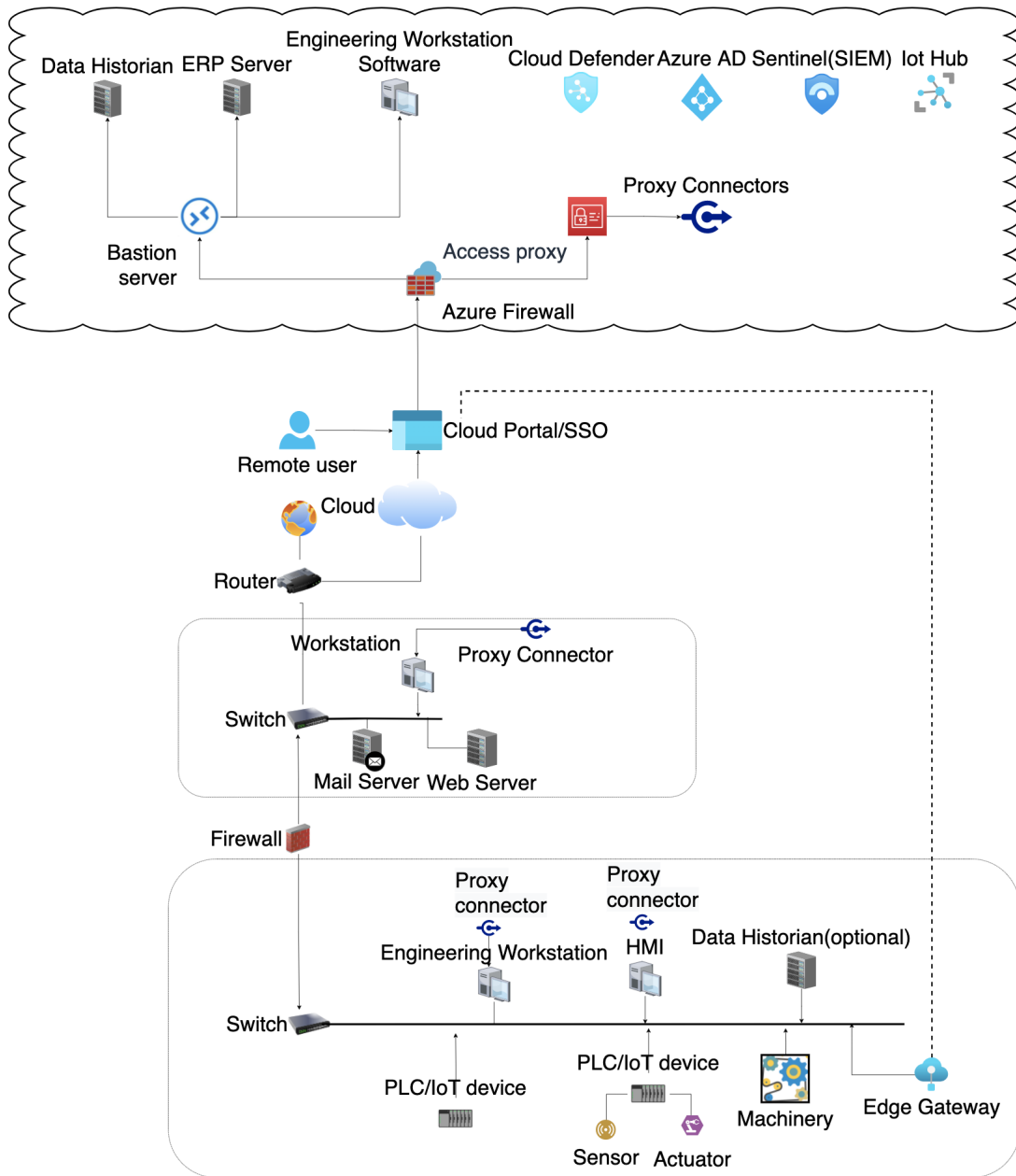


Figure 3.10: Cloud.

The biggest challenge in this situation is securing communications from and to the Cloud.

In this case, we are talking of Zero Trust Edge. Zero Trust edge, as we can read from the VMware site [71], it is a security solution that relies on the internet traffic to connect remote sites using Zero Trust access principles, primarily by utilizing cloud-based security and networking services.

To build a Zero Trust Edge Architecture, we need three main steps:

- Secure SD-WAN(software-defined Wide area network): this means to create

a secure connection between all Data centers, Saas, or services.

- Secure remote access: use cloud technologies to enable secure connection from remote to all users.
- Enable Zero Trust network access from anywhere: connected to the previous point, we need the capability to connect to the system regardless of location.

Essentially Zero Trust Edge is an extension of Zero Trust Architecture concepts with more attention to cloud solutions.

Applying Zero Trust concepts to Cloud is crucial for security, and many players have started to offer total cloud security solutions based on Zero Trust [10]. It becomes more and more clear that an infrastructure that requires interaction with cloud services cannot rely on perimeter-based security, even more than in an on-premises solution [20].

As we can read from the NIST publication [55] we fall in the use case of Multi-cloud/Cloud-to-Cloud solution. The NIST advice is to place a Policy Enforcement point (PEP) at the access point of each application or service, Cloud or local; in this way, we can still control all the accesses following the Zero Trust principles. To provide these functionalities, our solution is based on two main groups of components, those required for remote access and those required for Edge computing capabilities.

### 3.3.2 Remote access

In this cloud implementation, we hypothesize to have both on-prem devices and cloud-hosted VMs that require remote access from employees, so we need a secure way to allow remote access to these resources.

The adopted solution is based on two components that we have already used in the other two solutions:

- The Access proxy that, in this case, manages only the on-prem accesses providing RaaS (remote access as a service)
- The Bastion/Jump server that is used to access VMs in the cloud

In this case, both services are provided from the cloud, allowing to centralize the management of remote access easier. Even more important, the company does not need to worry about patches or maintenance. The service is continuously monitored to find vulnerabilities, and they are automatically patched with the latest updates.

## Access proxy

As in the case of the Zero Trust solution, a vital role is played by the Access proxy to control accesses. It is required to manage the connections to the different on-prem services. In this case, the access proxy works similarly to the on-prem ZTA one, but it is hosted on the cloud.

Clearly, we cannot avoid using a vendor solution for cloud implementation. Today the majority of vendors (Amazon, Azure, Google) provide services for the OT, and all of them offer more or less similar solutions.

For simplicity and because Microsoft, with Windows, is historically the leading supplier for this type of environment, for our solution, we took inspiration from the services offered by Azure.

Also, the cloud solution uses a SSO system for the authentication and authorization phase. A very useful and powerful solution is Azure Application Proxy [41], it is a service that became a part of the brand new “Microsoft Entra” family of services that uses Azure Active Domain (plays the role of Domain Controller), enabling users to access on-premises services from a remote client.

Application Proxy requires several components to work, and the two most important are the Application Proxy service and the Application Proxy connector:

- **Application Proxy service:** Runs in the cloud as part of Azure AD (Active Domain). It passes the sign-on token from the user to the Application Proxy Connector [40].
- **Application proxy connector:** A connector is a lightweight agent that runs on a Windows machine. It manages the communications between the Application Proxy service in the cloud and the on-premises application. The most important thing for security is that connectors only use outbound connections, so we do not have to open any inbound ports or put anything in the DMZ.

The connectors are stateless and pull information from the cloud as necessary. We can also implement more than one connector per service to implement load balance functionalities.

So we can leverage all the potentiality in terms of scalability, reliability, and also security of the cloud, also making a better user experience.

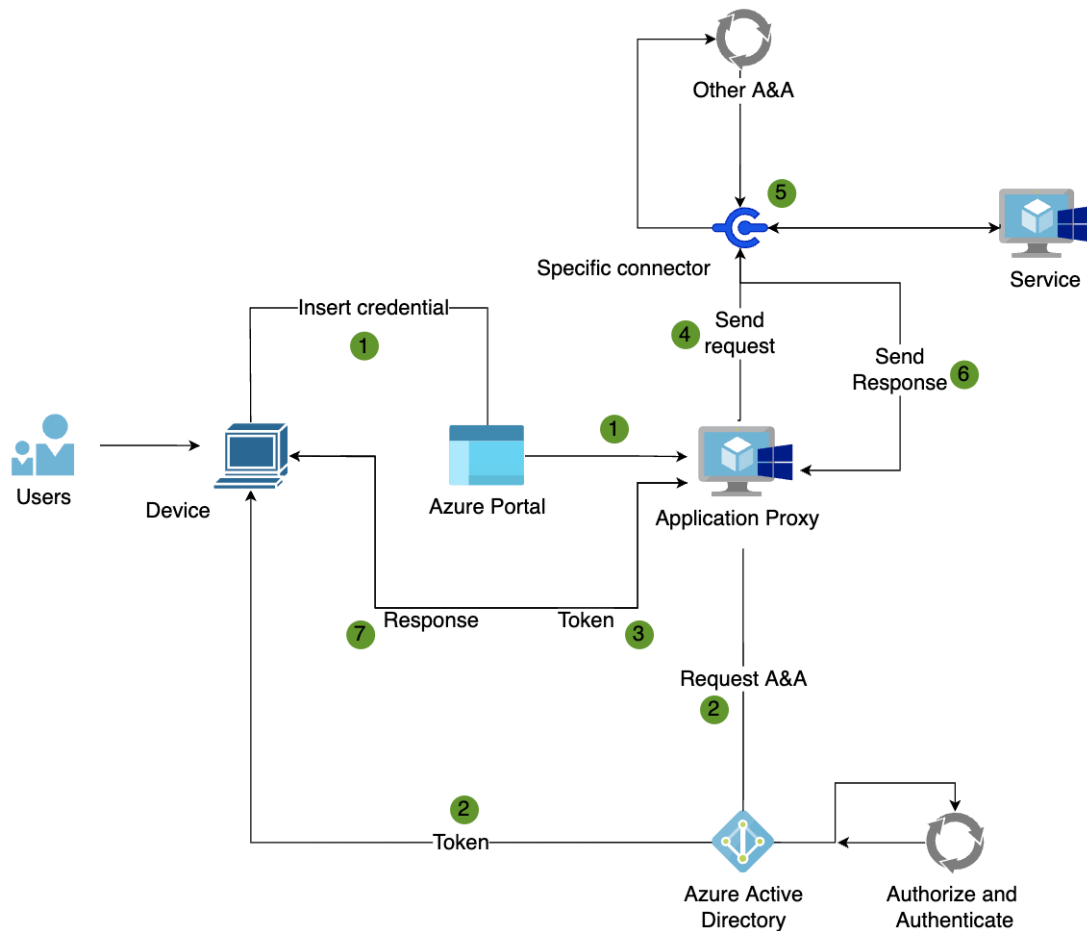
The service works in this way as also described in Figure 3.11:

1. After the user has accessed the application through an endpoint, he is redirected to the Azure AD sign-in page. We can configure Conditional

Access policies; in this case, specific conditions are checked at this time to ensure that the user complies with the organization's security requirements like level of privilege, the device from which the connection start, hour of connection, etc.

2. After a successful sign-in, Azure AD sends a token to the user's client device.
3. The client's device sends the token to the Application Proxy service, which retrieves the credentials from the token.
4. Application Proxy forwards the request, which is picked up by the Application Proxy connector
5. The connector performs any additional authentication required on behalf of the user, requests the internal endpoint of the application server, and sends the request to the on-premises application.
6. The response from the application server is sent through the connector to the Application Proxy service.
7. The response is sent from the Application Proxy service to the user.

In this way, we can authorize and authenticate any user following Zero Trust principles.



**Figure 3.11:** Cloud Authentication and Authorization flow.

In our case, the implementation is quite simple; we have in particular two devices that may require a remote connection, the HMI and the Engineering workstation.

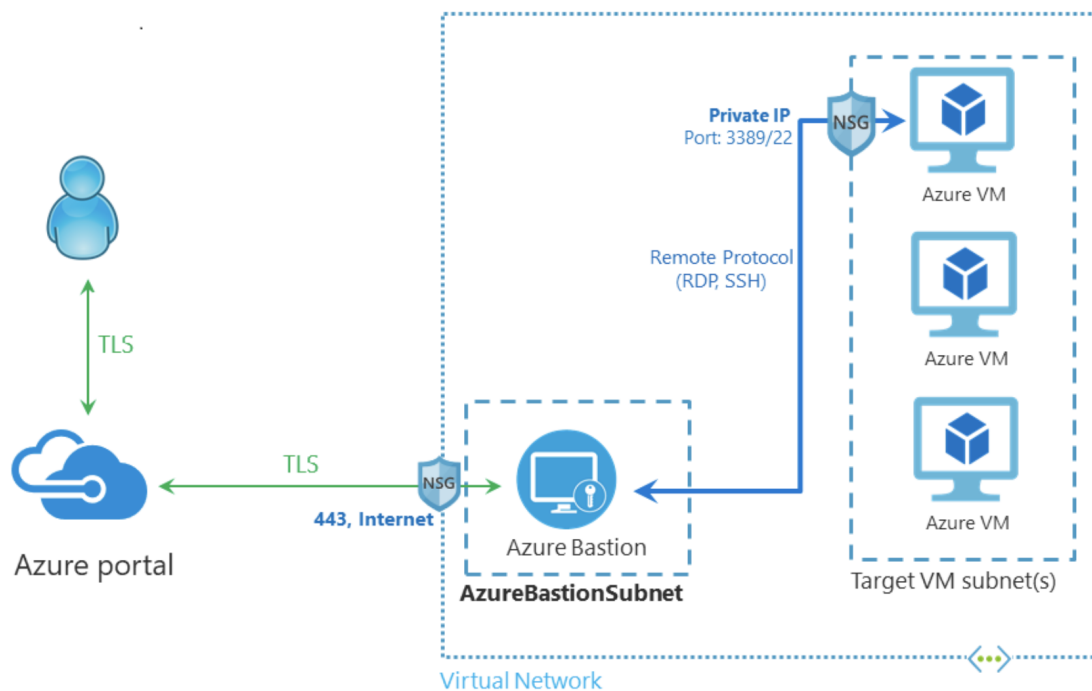
So, we have one connector for each device; in this way, whenever an employee needs to interact remotely with one of these devices, he can connect directly to the access proxy.

An essential point of this cloud service is that the ‘direct’ traffic of the user is terminated in the cloud. The Access proxy is, in fact, a reverse proxy that decouples the connection between the user and the back-end services. Furthermore, the connectors only use outbound connections to the Azure AD Application Proxy service, so there is no need to open firewall ports for incoming connections.

### Bastion/Jump Box

The Bastion service is the cloud equivalent of the jump server used in the first architecture (I-DMZ). Almost every cloud provider has its own implementation of

a Bastion service, an example is the Azure Bastion service that allows to access virtual machines through the internet and the Azure portal, providing SSH/RDP connection secured with the use of TLS protocol, it supports connection only from the TCP port 443.



**Figure 3.12:** Azure Bastion.

The service, described in Figure 3.12, allows to avoid exposing over the public internet SSH or RDP port of Virtual machines that, therefore, does not require a public IP address because is the Bastion that connects with the VMs through their private IP address. In this way we can apply the ‘least privilege principle’ restricting access to the VMs allowing connections only with Bastion. It isn’t a VM with a service installed but a PaaS service fully managed from the cloud provider, this means that does not require to be patched or updated.

Another advantage is that the service can be easily integrated with other security services like cloud firewalls or in the Azure case, with Microsoft Threat Intelligence, Azure Cloud Defender or Azure Sentinel (SIEM). In our infrastructure the service is placed behind the Cloud Firewall, in this way we have another layer of security between the internet and the cloud.

### 3.3.3 Edge Gateway

As said, another important component in our solution is the edge gateway. It is placed in the middle between the cloud and on-premises solutions and performs computation, aggregation, and several AI/ML analyses over the data before these are sent to the cloud to reduce bandwidth usage and increase the efficiency of the systems.

There are several ways to implement this type of device. However, also, in this case, Azure (like the other platforms) offers several services not only for implementing these functionalities, but also to integrate the on-prem or proprietary solution with the cloud, to support the company in the transitions as smoothly as possible. An example is the service called “**Azure IoT Edge**” [49], a very flexible service that offers a huge number of functionalities. It is based on Docker containers to run the business logic at the edge. It integrates a lot of event processing, machine learning, image recognition, and other AI functionalities but also allows the integration of proprietary code and services. The service is composed of three main components:

- **IoT Edge Cloud Interface:** Provide a panel from which to manage and monitor all IoT devices and create and configure workloads.
- **IoT Edge modules:** These execution units can be implemented as a Docker-compatible container that runs business logic at the edge. Multiple modules can be configured to communicate, creating a data processing pipeline.
- **IoT Edge Runtime:** Enables custom and cloud logic on IoT Edge devices. The runtime sits on the IoT Edge device and performs management and communication operations.

One of the most important characteristics of this type of service is flexibility. In the last years, one of the main problems that have been holding companies back from the cloud transition is the compatibility of old or proprietary technologies on which are based the infrastructures. With services like Azure IoT Edge, we can connect these technologies with the cloud with minimum effort. It supports two main protocols, MQTT and AMQP, but also provides a way to translate from any other protocol (Modbus, BLE, BACnet, OPC-UA). This capability can be enabled by using an IoT Edge device as a gateway [42] providing connection between the IoT Hub and devices.

In particular, we have two different deployment models for the Gateway [44]



- **Transparent Gateway:**
- In the transparent gateway pattern, devices that theoretically could connect to IoT Hub are connected to a gateway device instead. The downstream devices have their own IoT Hub identities and connect using MQTT or AMQP protocols. The gateway passes communications between the devices and IoT Hub. The devices and the users interacting with them through IoT Hub are unaware that a gateway is mediating their communications. This lack of awareness means the gateway is considered transparent. In this case, we have a so-called Parent/Child relationship between the devices (PLCs or other IoT devices) and the gateway. From the security point of view, the Zero Trust principle is respected; in fact, we have that Childs and Parent devices need to authenticate their connection to each other through the use of X.509 certificates.
- **Translation gateway:** This deployment model is used when devices use a different protocol from the ones used by Azure IoT (MQTT and AMQP); it is up to the gateway to translate from the protocol used by the devices to the ones supported. We can extend the functionalities of the gateway by installing custom or third-parties modules specific to the downstream protocol used by devices.

For this model, we also have two different possibilities:

- **Protocol translation:** In the protocol translation gateway pattern, only the IoT Edge gateway has an identity with IoT Hub. The translation module receives messages from downstream devices, translates them into a supported protocol, and then the IoT Edge device sends the messages on behalf of the downstream devices. All information looks like it is coming from one device, the gateway. Downstream devices must embed additional identifying information in their messages if cloud applications want to analyze the data on a per-device basis.
- **Identity translation:** The identity translation gateway pattern is built on protocol translation, but the IoT Edge gateway also provides an IoT Hub device identity on behalf of the downstream devices. The translation module is responsible for understanding the protocol used by the downstream devices, providing them identity, and translating their messages into IoT Hub primitives. Downstream devices appear in IoT Hub as first-class devices. A user can interact with the devices

in IoT Hub and is unaware of the intermediate gateway device.

Every IoT hub maintains an identity registry that stores information about the devices and modules permitted to connect to it [43]. If a device wants to connect to the IoT Hub, there must be an entry for that device in the IoT hub's identity registry.

The Identity registry is another important component; it is the equivalent for IoT devices of the Azure AD service; it stores the device identities. The authentication of the devices is done using an X.509 certificate that allows authentication of an IoT device at the physical layer as part of the Transport Layer Security (TLS) standard connection establishment.

So once the device is authenticated, the communication is protected by the TLS protocol.

In our solution, we use the transparent deployment model if possible. The Identity Translation solution has been chosen if it is not possible because some devices could require a translation. This choice is because these two solutions allow them to interact directly with the devices, providing more fine-grained control and visibility, allowing for better monitoring of the overall architecture, with no drawbacks in terms of security.

The Edge Gateway is placed in the I-DMZ to avoid direct connections to the OT. All the communications of PLCs and any devices from the OT that require sending data to the cloud must pass through the OT/I-DMZ firewall and the gateway providing another layer of security. Also in this case, the advantage of the cloud is clear:

- We do not need any direct connection to machines or PLCs; we can upload workloads on the devices from the cloud.
- We can easily centralize logs and report of any devices in the cloud
- We do not need to care about patches or updates of the services
- All communications with the external world are intermediate by the Edge gateway.

### 3.3.4 Data Historian

As we can see in the other two architectures, an essential component in an industrial environment is the Data historian. If we look at what the market offers, any cloud provider offers a solution to implement memorization. Cloud

technologies enable different functionalities as-a-Service helpful for monitoring the production that otherwise should be implemented manually with significant use of company resources.

Any cloud provider offers a solution to implement a database system, which is what essentially is a Data Historian, so a company can implement the Data Historian on the cloud based on their needs without forgetting to take into account the “vendor lock-in” problem. In the cloud, we can easily implement, in a pay-per-use way, essential services like predictive maintenance or ML (Machine Learning) solutions for the analysis of trends to have better insight into the production.

Because of the proliferation of IoT devices, more storage capabilities are needed, and the use of the cloud is an obvious choice to solve the problem.

Day by day, industrial devices generate billions of data that require to be processed and stored, this, in an on-prem solution, could become a problem because it is required a continuous update of the storage capabilities. So, another problem that the cloud can help to solve is storage capabilities. Resources can scale with our needs in a transparent and on-demand way without any effort. To connect PLCs and other industrial devices to the cloud-hosted Historian, we can use the Access proxy service and connectors to securely send data to the cloud.

Another widespread solution is to have an on-prem historian placed in OT that sends data to the Edge gateway and, after pre-processing, to the cloud, to extract useful information from the vast amount of generated data. This solution could be suitable in the early stages of a cloud transition. Data historian contains essential information, crucial for the production, but often could also contain, as previously said, industrial secrets, and this is because a company could be reluctant to move the Historian to the cloud.

With this solution, we maintain our data inside the on-prem Historian, thus being able to decide which ones we want to send to the cloud for processing. Moving the Historian to the cloud is another important step in the cloud transition that provides many benefits for the business, enabling data mining capabilities to extract much more information from the data and removing the problem of machine and storage management.

### **3.3.5 Security services**

From the security point of view, Cloud offers a large number of services, the continuous monitoring capabilities that are embedded in a cloud solution and the ease with which the infrastructure can be kept up to date offer better basic

security in comparison with an on-prem solution. Examples of security services offered by Microsoft are ‘Microsoft defender’ [45], ‘Microsoft Threat Intelligence’ or Microsoft Sentinel. Microsoft Defender for Cloud is a CSPM (Cloud Security Posture Management), so it plays a role in providing information, alerts, hardening recommendations, vulnerability assessment, and assets inventory to secure, step-by-step, the infrastructure.

One of the most important features of Microsoft Defender is that it is compatible also with on-premises devices and cloud resources from other vendors, thus limiting one of the main drawbacks of the Cloud, which is vendor lock-in. The service can be associated with any type of resource (Networks, storage, etc.) in the network and, thanks to the Microsoft Threat Intelligence insight, can provide visibility and security for any of them. As said, it is strictly connected with the service of Threat Intelligence that provides information about the newest vulnerabilities. A service of Threat intelligence [47] is crucial and goes beyond lists of bad domains or bad hashes. Instead, it provides the necessary context, relevance, and priority for people to make faster, better, and more proactive cybersecurity decisions.

Another Service is Microsoft Sentinel, as other security services that can be connected with all the services and provide the functionalities of a SIEM but hosted on the Cloud.

### 3.3.6 Cloud summary

Cloud brings many benefits not only from the security point of view but also in terms of functionalities, flexibility, and reliability with respect to on-prem solutions. In the IT world, even if there is still skepticism towards the cloud, it is widespread, in OT, we are still in the infancy of cloud adoption. In a context where there are Industrial secrets and where the concept of “Security by obscurity” was predominant, moving to the cloud is a challenge, and the fear of losing control on data is still relevant in the manager’s choice. Moving to the cloud is an important choice; it is a process that requires planning, time, competencies, money, and several steps. It is important to understand well the company’s business goals and the maturity level to define a strategy for the transition. There could be cases where moving to a cloud solution is not the best choice, but the trend is clear, and the competitive advantage that gives the cloud is unavoidable.

# Implementation

In this thesis, we have proposed possible architectures based on the I-DMZ and Zero Trust. In particular, we want to highlight how a Zero Trust model can be an essential ally for the security of Industrial infrastructures, to secure remote accesses and all the interactions between the different components of the architecture. The solution is identity-centric and allows monitoring of any access to any resources, taking into account the context in which this interaction takes place. The core of the solution is the authorization and authentication phase done through a trust algorithm that uses several information from different sources to define the context of the communication to understand if it can be authorized or not.

The proposed solution is based on two main components the Access proxy, which deals with the authorization and authentication (A&A) of all connections, and the Next Generation Firewall, located at the heart of architecture, managing and monitoring all the communication between the different components. Because of that, the implementation and validation phases are focused on these two components.

First, we have implemented the access proxy that uses a trust algorithm to authorize and authenticate all the users that try to connect to the company, testing the resiliency of this solution and the ability to recognize a malicious login attempt. For the NGFW, we tested its ability to dynamically adapt to the specific context, managing the connections inside the network.

## 4.1 Implementation Access Proxy

The core of the access proxy is the Trust algorithm that defines who can access which service and with which privilege level. In a production implementation, the trust algorithm can be made more or less complex according to different factors and requirements, adding new features and factors for authenticating and authorizing the user. In our case, the scope is to determine how a Zero Trust

approach can make it more difficult to bypass the A&A phase. We assume that a malicious actor managed to steal an employee's credentials and try to use them to access remotely to the company network. In this case, our approach can block this attack, detecting the intrusion attempt and blocking it. The language that we have used is Python, one of the most powerful languages that, thanks to the high number of libraries and extensions, allows us to develop complex solutions in a fast and easy way. We have used Flask <sup>1</sup> to simulate the Single-Sign-On system of the company's network with a login page and the back-end for the authorization and authentication.

As described in the Zero Trust part of this thesis, the authorization and authentication phase is based not only on the user's information but also on the device, from which the user is trying to connect to the company's network. In this way, even if an attacker was able to steal the credential of a user, the access proxy can identify the device and understand if it is the right one. The Domain controller plays an important role in this process, which stores all the information about the users, their role, and their level of privilege.

In our solution, the role of the DC has been simplified by using a MySQL server. To implement the needed functionalities in Python, we have used different libraries:

- **PyJWT:**

This is the library that allows to manage of JSON Web Tokens (JWT) <sup>2</sup>. A JWT is an open standard described in the RFC 7519 that defines a format to securely encode and transmit JSON objects between parties. A JWT can also be encrypted using a secret with the HMAC algorithm or with a public/private key pair using RSA or SHA256. It is widely used in Single-Sign-on systems because of its small overhead and its ability to be easily used across different domains. Once the user is logged in, we generate the JWT with the needed information about the user and the device, and from this moment, each subsequent request will include the token. In this way, we can identify and authorize the user every time he requests and modify, for example, the level of privilege of the specific user accordingly to the context. A possible scenario is one in which an employee is trying to access the network from an unknown device. In this case, the system administrator may decide to deny the access at all, or he may decide to allow restricted access to a limited number of services (maybe he can block

---

<sup>1</sup>Flask[21] is a framework for Python that allow defining in an easy and fast way web applications.

<sup>2</sup>Bradley J. Jones M. and N. Sakimura. JSON Web Token JWT RFC 7519. May 2015. url: <https://www.rfc-editor.org/info/rfc7519>.

the access to the OT but leave the access to the IT), accordingly with the level of Risk associated with each resource.

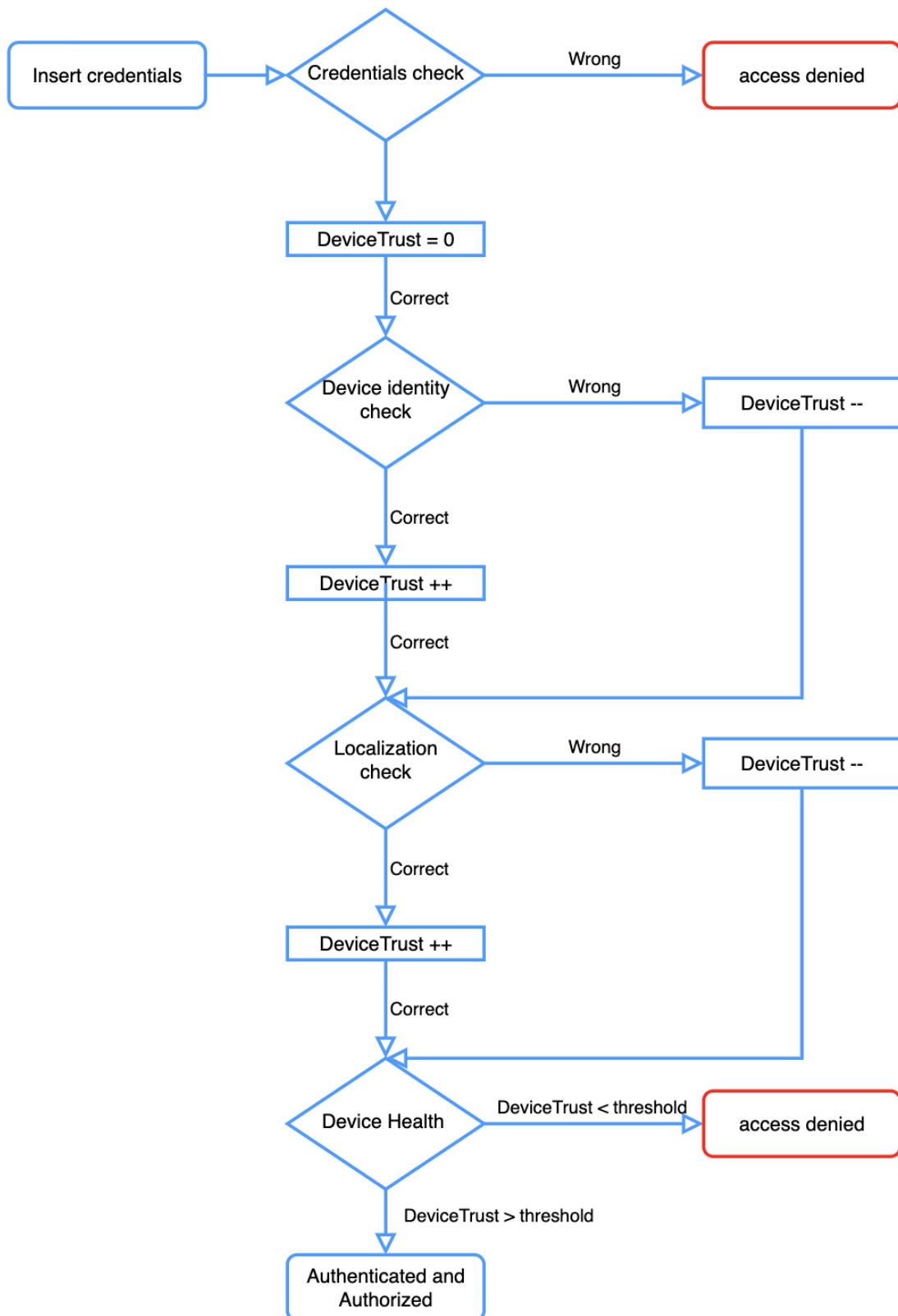
- **Nmap**: Another tool that we use is Nmap[56]; it is one of the most famous open-source tools for port scanning, network discovery, and security auditing. In our case, it is used to extract more information about the device from which the user is trying to connect to the company network. In particular, it is used to determine the type of device, the Operative system, and its version. This information is useful not only to determine if the device is the one associated with the specific user, but it may also be used to better define the context and the Risk level of the connection.

An example may be the case where the user's device, despite being the associated one to the specific user, has not been updated with the latest release, and the installed version has some unpatched vulnerabilities. Also in this case, the system administrator can decide to block the user's access, even if the device is the right one, forcing to update the device. As previously said, we can create fine-grained access policies accordingly to the needs of the specific infrastructure or context. The use of Nmap and port scanners, in general, is controversial, as highlighted in the official page [57], it can be considered legal or illegal based on the country we are in.

In our case, the problem does not exist because we are scanning the device of our employees, and we can get them to sign a statement of consent.

### 4.1.1 Algorithm

As said, the algorithm is implemented in Python and is composed of several steps of authentication and authorization described in Figure 4.1.



**Figure 4.1:** Trust algorithm Access proxy.

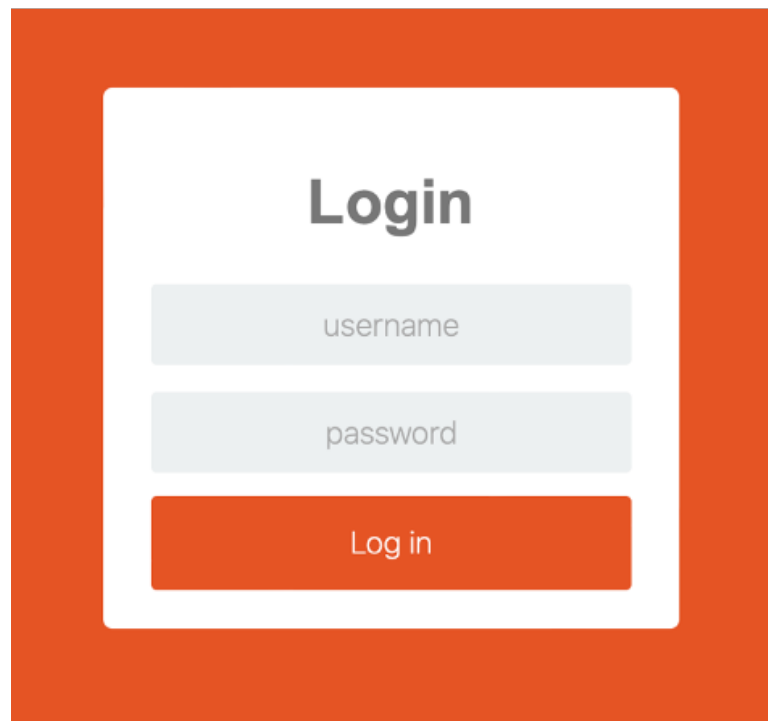
The Trust algorithm is fully customizable accordingly to the specific policies



and procedures adopted by the company. At each step, we modify the level of trust given to the device based on different indicators (device identity, localization of the connection, device health). Once that the algorithm is completed, we can decide whether to deny access or to grant it with limited privileges, based on a threshold.

### Verification of the credentials

First of all, the user inserts the company's credentials on the login page.



**Figure 4.2:** Login panel.

Once the user has entered the credentials, the system sends a query to the domain controller to verify that the user is in the system, decrypt and check the password entered.

### Device identification

At this point, we start to identify the device and all the information that we need.

```
if passwordCheck:
    #identify the ip address and the mac address of the remote user
    remoteIp = request.remote_addr
    remoteMac = get_mac_address(ip=remoteIp)
    #use nmap to identify the device information
    nm.scan(remoteIp, arguments="-O")
    remotrDeviceType = nm[remoteIp]['osmatch'][0]['osclass'][0]['type']
    remoteOs = nm[remoteIp]['osmatch'][0]['osclass'][0]['osfamily']
    remoteOsVersion = nm[remoteIp]['osmatch'][0]['osclass'][0]['osgen']

    if(remoteMac == 'None'):
        remoteMac = nm[remoteIp]['addresses']
```

Figure 4.3: Device identification.

As we can see from the image ?? we retrieve the IP address of the device and, from this, we try to extract the MAC address with the module `get_mac_address()` of the package `getmac`<sup>3</sup>. The module provides a platform-independent interface to get the MAC addresses of system network interfaces (by interface name) and remote hosts on the local network (by IPv4/IPv6 address or hostname).

The function uses the ARP protocol to retrieve the MAC address from the IP. This means we can use it to identify hosts in our local layer 2 networks. This is the reason why the last "if" in the image ??.

If the host is in the same network of the Access proxy, we can use the result of `get_mac_address()`, if instead, the host is not in the same LAN, we use the value retrieved by nmap. At this point, we use Nmap with the "-O" flag that which stands for "OS detection" to retrieve other information about the host, this command gets a lot of insight that, as described in the official documentation[58] can be used for several scopes like:

- Determining vulnerabilities of target hosts
- Network inventory and support
- Detecting unauthorized and dangerous devices (our goal)

In our case, we store just a few information:

- **remoteDeviceType:** is used to determine if the device is a PC, a smart-phone, or any other type of machine.

<sup>3</sup>getmac url: <https://pypi.org/project/getmac/>

- **remoteOS:** identify the OS system used from the host
- **remoteOsVersion:** identify the version of the OS

This information is useful both to identify the device and to better delineate the context and the level of risk of the connection. As previously said, we can modify our security posture based on the context in which the connection occurs.

Suppose we know, through our threat intelligence system, that a specific version of an operative system is affected by a vulnerability. In that case, we can limit the privilege of that specific host forcing the employee to update it.

### Vulnerabilities control

To determine if the specific device is affected by some vulnerabilities, we use two important public-available databases.

The first DB that we need is provided and maintained by the Cybersecurity & Infrastructure Security Agency (CISA) and is the *"Known Exploited Vulnerabilities Catalog"* [7] that contains all the vulnerabilities discovered so far.

The second DB is the *"National Vulnerability Database"*, provided by the NIST. The agency exposes an API [54] through which we can retrieve all the information regarding a specific Common Vulnerabilities and Exposure (CVE) vulnerability. The CVE [51] program is an important international project overseen by the MITRE corporation with funding from the CISA and part of the U.S. Department of Homeland Security, which aims to offer a global and centralized list of publicly disclosed computer security flaws.

An important value that we retrieve from these DBs is the *"baseScore"* of a vulnerability. The base score value is a vulnerability metric that represents the innate characteristics of each vulnerability, it ranging from 0 to 10, and is defined from the Common Vulnerability Scoring System (CVSS) framework [53], it is a function of the Impact and Exploitability scores:

- **Exploitability score:** the exploitability sub-score represents metrics for Attack Vector, Attack Complexity, Privileges Required, User Interaction, and Scope. The sub-score measures how the vulnerability can be accessed, the complexity of the attack, any required privileges, the interaction needed between the attacker and another user, and the impact on resources beyond the vulnerable component.
- **Impact score:** The impact sub-score represents metrics for confidentiality impact, integrity impact, and the availability impact of a successfully

exploited vulnerability.

```
def get_device_risk(device, deviceVersion):
    print(deviceVersion)
    baseScoreThreshold = 8.0
    vulnerable = False
    # Creating request object to KNOWN EXPLOITED VULNERABILITIES CATALOG
    req = urllib.request.Request(CISA_VULN)
    # Getting in response JSON
    response = urllib.request.urlopen(req).read()
    json_response = json.loads(response.decode('utf-8'))
```

Figure 4.4: Set Thresholds.

As we can see in the image 4.4 We set a *baseScoreThreshold* and create a request object using the `urllib.request` module that returns a JSON document with all the vulnerabilities inside the CISA DB. Once we have the list, we can iterate to determine if our device is affected by some vulnerability and if so, we print the name of the vulnerability and save the CVE ID.

As we can see in the image 4.5, we can use the NIST API to retrieve information about the CVE.

```
# Creating request object to NATIONAL VULNERABILITY DATABASE and add the specific CVE
req = urllib.request.Request(CVE_INFO+cve)
# Getting in response JSON
response = urllib.request.urlopen(req).read()
json_response2 = json.loads(response.decode('utf-8'))
baseScore = json_response2['result']['CVE_Items'][0]['impact']['baseMetricV3']['cvssV3']['baseScore']
attackVector = json_response2['result']['CVE_Items'][0]['impact']['baseMetricV3']['cvssV3']['attackVector']
exploitabilityScore = json_response2['result']['CVE_Items'][0]['impact']['baseMetricV3']['exploitabilityScore']
impactScore = json_response2['result']['CVE_Items'][0]['impact']['baseMetricV3']['impactScore']
print("Base score: ", baseScore)
print("exploitability score: ", exploitabilityScore)
print("impact score: ", impactScore)
print("attack vector", attackVector)
if attackVector == "PHYSICAL":
    deviceTrust += (impactScore + exploitabilityScore) - 1
elif attackVector == "LOCAL":
    deviceTrust += (impactScore + exploitabilityScore) - 2
elif attackVector == "ADJACENT":
    deviceTrust += (impactScore + exploitabilityScore) - 3
elif attackVector == "NETWORK":
    deviceTrust += (impactScore + exploitabilityScore) - 4
```

Figure 4.5: CVE request.

The API returns helpful information that can be used to set more fine-grained controls. In our case, we save the `baseScore`, the `exploitabilityScore`, the `impactScore`, and the `AttackVector`. As shown in Figure 4.5, even if the `AttackVector`

value is included in the calculation of the impact score, we want to give more emphasis to this value. This is because, in evaluating the risk associated with remote access, this value reflects the context by which vulnerability exploitation is possible; in particular, this metric value will be larger the more remote an attacker can be in order to exploit the vulnerable component. Another parameter that we retrieve from the NIST DB is the **"cpe\_match"** that contains CPE information<sup>4</sup>.

```
#control if the specific version is affected from the vulnerability
for productVersion in json_response2['result']['CVE_Items'][0]['configurations']['nodes']:
    for cpe in productVersion['cpe_match']:
        if deviceVersion in cpe['cpe23Uri']:
            print("your device version is not patched for the vulnerability: ", cve)
            deviceTrust -= 2
            break
        else:
            print("your device version is patched for the vulnerability: ", cve)
            deviceTrust += 2
            break
deviceTrust = deviceTrust / (counter)
print("The final trust level is", deviceTrust)
return deviceTrust
```

**Figure 4.6:** Control vulnerable version.

As we can see in the image 4.6, we iterate over the **"cpe23uri"** element trying to determine if the specific version of our device is affected or not by the specific vulnerability that we have found. Whether the specific version is patched or not is influenced by the device's trust level. Finally, we divide the device trust score by the number of vulnerabilities we found. In this way, the number of vulnerabilities will negatively affect the trust given to the device.

## Localization

Once we have identified and checked the information about the user and the device, we want other information to identify from which location the request came.

To do that we have implemented the function `get_location()`.

<sup>4</sup>The Common Platform Enumeration (CPE) [50] is another open Framework that provides a structured naming scheme for information technology systems, software, and packages, the MITRE corporation oversees it and currently maintained by the National Institute of Standards and Technology (NIST) as part of its U.S. National Vulnerability Database (NVD).

```
def get_location():
    # Creating request object to GeoLocation API
    req = urllib.request.Request(GEO_IP_API_URL+IP_TO_SEARCH)
    # Getting in response JSON
    response = urllib.request.urlopen(req).read()
    # Loading JSON from text to object
    json_response = json.loads(response.decode('utf-8'))
    # Print country
    return json_response
```

Figure 4.7: Localization.

The function use the `urllib.request` module<sup>[63]</sup> to contact the `ip-api` service<sup>[2]</sup>.

The service exposes an API that retrieves much useful information from the IP-address, as we can see from the following image 4.8.

```
"query": "137.204.24.147",
"status": "success",
"continent": "Europe",
"continentCode": "EU",
"country": "Italy",
"countryCode": "IT",
"region": "45",
"regionName": "Emilia-Romagna",
"city": "Bologna",
"district": "",
"zip": "40125",
"lat": 44.488,
"lon": 11.3752,
"timezone": "Europe/Rome",
"offset": 7200,
"currency": "EUR",
"isp": "BOLOGNA-ALMA",
"org": "Alma Mater Studiorum Universita' di Bologna",
"as": "AS137 Consortium GARR",
"asname": "ASGARR",
"mobile": false,
"proxy": false,
"hosting": false
```

Figure 4.8: ip-api info.

So we store this information to compare them with those associated with the user. In particular, we use the information about the country, the region, and the city as we can see from the image 4.9.

```
if(macData == remoteMac and ipData == remoteIp and osData == remoteOs):
    # identify from which location the request come from
    location = get_location()
    country = location['countryCode']
    region = location['regionName']
    city = location['city']
```

Figure 4.9: Verify device.

In this case, we use just a small number of the information retrieved, but in another context, we can use more of it to set more fine-grained authorization policies. Also the information about the location from which the connection comes will affect the device trust level.

## Login

Finally, we compare the device trust score with a given threshold that can be dynamically modified according to the context and resource the user is trying to access. We authorize the user if everything is fine and the device's trust score is high enough. At this point, we use the JWT module to generate the token, which will be associated with the user's traffic to trace all the operations.

The token is also encrypted, in this case with the **HMAC SHA-256** algorithm and a randomly generated key.

```
if(countryData == country and regionData == region and cityData == city):
    token = jwt.encode({'public_id': userName, 'mac': macData, 'ip': remoteIp, 'country': country,
                       'region': region, 'city': city, 'os_version': remoteOsVersion, 'device_type': remotrDeviceType,
                       'exp': datetime.datetime.utcnow() + datetime.timedelta(minutes=45)}, app.config['SECRET_KEY'], "HS256")
    session['logged_in'] = True
    return home()
```

Figure 4.10: Verify location.

This solution allows, in a simple way, to implement a Zero Trust approach to manage the access to an Industrial infrastructure (or any other type of infrastructure).

One of the greatest strengths of this approach is the flexibility; we can extend it based on our needs and tailor the authentication and authorization process to the specific infrastructure.

## 4.2 Implementation Next Generation Firewall

The second most crucial component of our Zero Trust solution is the Next Generation Firewall that is placed in the middle of the architecture between IT, OT, and the Access control part. All the connections are forced to pass through the NGFW, which is responsible for monitoring and controlling the traffic between the different parts of the network.

To implement this part of the architecture, we have used VMware [70] and an Ubuntu VM that acts as the NGFW and proxy to analyze, route, and filter the communications.

### 4.2.1 Firewall

An Ubuntu VM plays the role of the NGFW and is placed in the middle between the PLC (OT) and the host machine that plays the role of the IT machine, which will be better described later. Because all the communications pass through the NGFW, it also acts like a router allowing communications between the two networks.

The routing functionalities are activated through the command `sysctl -w net.ipv4.ip_forward=1`, that changes the kernel parameter. The Firewall functionalities are implemented with Iptables as described in the following section.

#### Iptables

Iptables [52] is a software that allows the implementation of the functionality of Packet Filtering (both static and dynamic), Inspection, NAT, and package marking (mangling). It is installed in all major Linux distributions since Kernel 2.4, and it is a direct descendent of Ipchains (Kernel = 2.2).

Iptables allow us to define rules for packet filtering, it is interfaced with the Linux kernel via the Netfilter module, which operates by providing hooks to the operating system, which can be used to intercept packets in transit in the TCP/IP kernel stack.

The rules defined with Iptables allow us to implement management functions associated with a specific hook. Each time a packet passes through a hook, Netfilter checks if it has been assigned a management function. In case of a match, the packet is passed to the function, otherwise the packet switches to the next hook. Hooks:

- **NF IP PREROUTING:** is reached by incoming packets through a network



interface and before being routed.

- **NF IP LOCAL IN:** is reached only by packets directed to the local machine
- **NF IP FORWARD:** is reached only by packets coming from a network interface and directed to another interface (packets in transit)
- **NF IP LOCAL OUT:** is traversed by locally generated packets before being subjected to the routing logic.

For each packet that passes through a hook, we can manage them with different actions.

- **ACCEPT:** accepts the packet that is passed to the next hook.
- **DROP:** deny access to the package, the packet is not passed to the next hook, all resources related to the package are deallocated, and the packet is discarded.
- **QUEUE:** queues the packet that is made available for management in user space
- **NFQUEUE:** This target is an extension of the QUEUE target. As opposed to QUEUE, it allows us to put a packet into any specific queue, identified by its 16-bit queue number. `-queue-num value`. This specifies the QUEUE number to use.
- **RETURN:** means stop traversing this chain and resume at the next rule in the previous (calling) chain.

If the end of a built-in chain is reached or a rule in a built-in chain with target RETURN is matched, the target specified by the chain policy determines the packet's fate.

Iptables uses the concept of tables to organize rules. These tables classify the rules according to the decisions that can be taken within them (filter = allow a packet to pass, NAT = network address translation). Within each table, there is a number of built-in chains, it may also contain user-defined chains. These chains represent the Netfilter hooks that activate the different tables and the related actions.

The main tables<sup>[18]</sup> are:

- **filter:** This is the default table, it contains the built-in chains INPUT (for packets destined to local sockets), FORWARD (for packets being routed through the box), and OUTPUT (for locally generated packets).
- **nat:** This table is consulted when a packet that creates a new connection is encountered.  
It consists of three built-ins: PREROUTING (for altering packets as soon as they come in), OUTPUT (for altering locally-generated packets before routing), and POSTROUTING (for altering packets as they are about to go out).
- **mangle:** This table is used for specialized packet alteration. Until kernel 2.4.17, it had two built-in chains: PREROUTING (for altering incoming packets before routing) and OUTPUT (for altering locally-generated packets before routing). Since kernel 2.4.18, three other built-in chains are also supported: INPUT (for packets coming into the box itself), FORWARD (for altering packets being routed through the box), and POSTROUTING (for altering packets as they are about to go out).
- **raw:** This table is used mainly for configuring exemptions from connection tracking in combination with the NOTRACK target. It registers at the Netfilter hooks with higher priority and is thus called before ip\_conntrack, or any other IP tables. It provides the following built-in chains: PREROUTING (for packets arriving via any network interface) OUTPUT (for packets generated by local processes)

Iptables is a powerful tool that allows the configuration of fine-grained filters and policies to manage traffic through the machine.

### Application level analysis

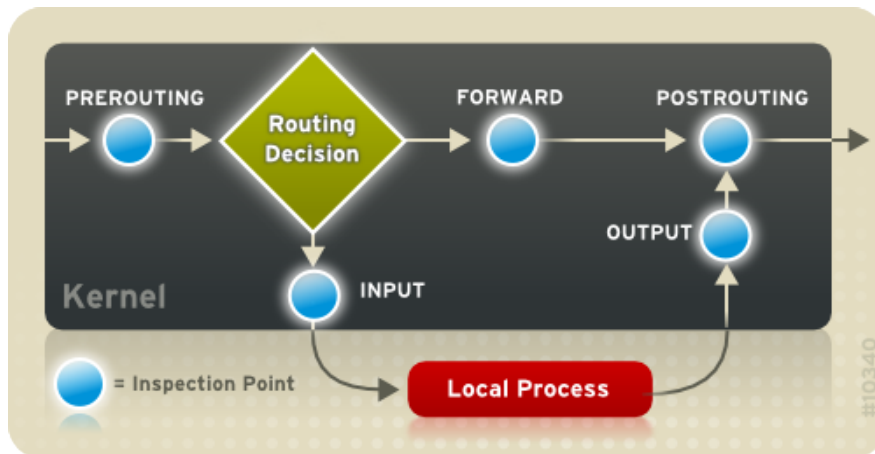
In our case, we exploited only a small part of the potentialities of Iptables. In particular, we have configured iptables to put the packets directed to the IP address of the PLC in the Forward chain and send them to the application level through queuing to analyze them and decide if drop it or let it pass.

The configuration command that we have used is the following:

- `iptables -I FORWARD -d 169.254.179.67 -j NFQUEUE --queue-num 1`
  - With this command, we are saying to Iptables to insert a new rule in the FORWARD chain (-I FORWARD)

- which has as destination IP the one of the PLC (-d 169.254.179.67)
- to the target (-j) NFQUEUE in the queue 1.

The image 4.11 shows how NFQUEUE works.



**Figure 4.11:** Netfilter packets flow.

Once we have captured the packet in the queue, thanks to the `NetfilterQueue`<sup>5</sup> wrapper for `libnetfilter_queue`<sup>6</sup>, we can manage it in the Python code. `NetfilterQueue` provides access to packets matched by an iptables rule in Linux. This way, matched packets can be accepted, dropped, altered, reordered, or given a mark.

In Figure 4.12 we can see the code for the binding with queue 1 of NETFILTER, to take the packet, that will be managed in the "drop\_or\_accept" method for the authorization phase.

```

nfqueue = NetfilterQueue()
nfqueue.bind(1, drop_or_accept)
try:
    nfqueue.run()
except KeyboardInterrupt:
    print('')
nfqueue.unbind()

```

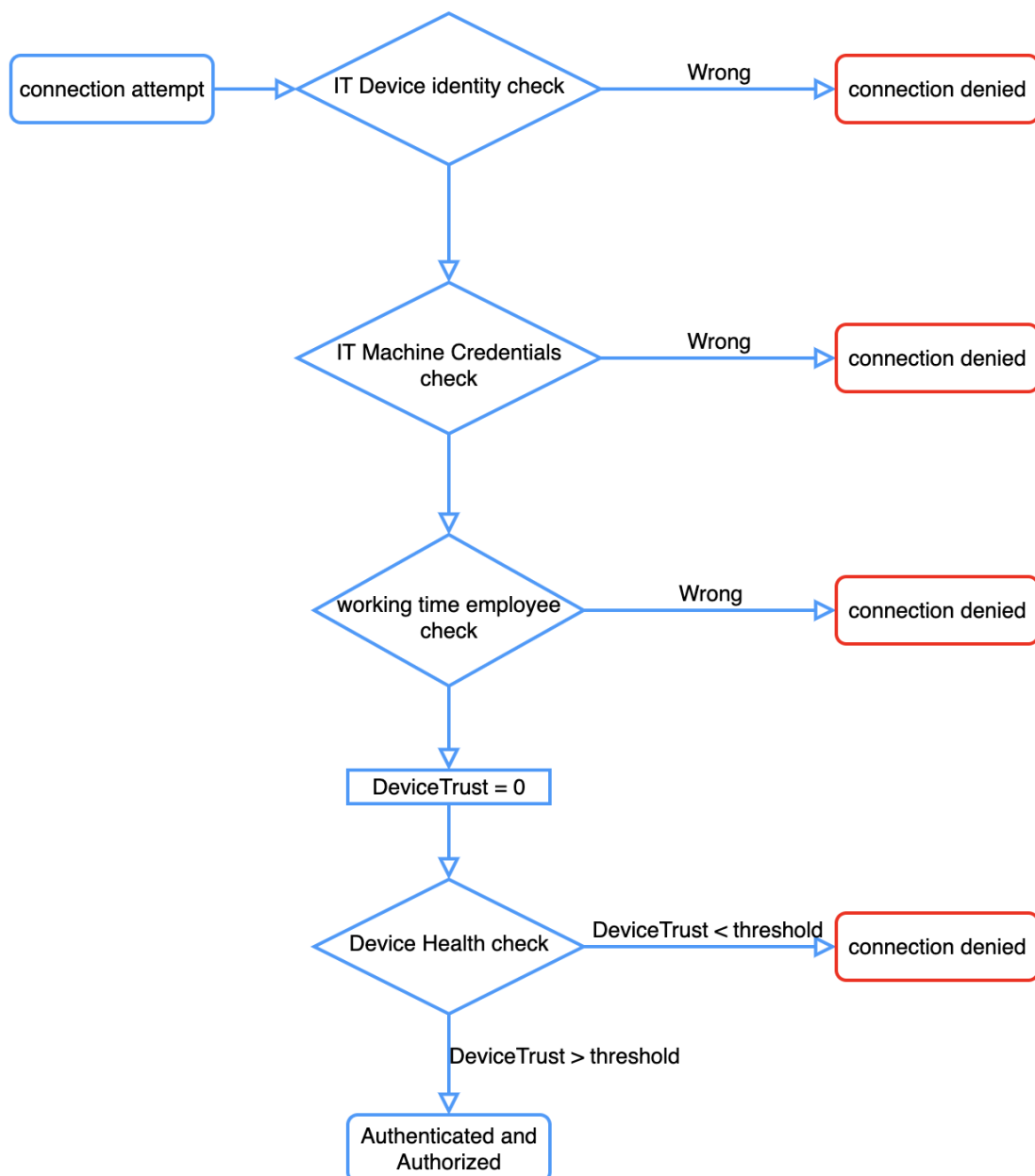
**Figure 4.12:** NetfilterQueue bind code.

<sup>5</sup>Netfilter url: <https://pypi.org/project/NetfilterQueue/>

<sup>6</sup>Netfilter project url: [https://www.netfilter.org/projects/libnetfilter\\_queue/](https://www.netfilter.org/projects/libnetfilter_queue/)

## Authorization

Once we have the packet, we want to inspect it to determine the device's trust level, from where it comes, and if it is authorized to reach the device in the OT. As in the case of the Access proxy, the NGFW needs to define the trust score of the connection, and to do that, it runs a trust algorithm as described in Figure 4.13.



**Figure 4.13:** Trust algorithm NGFW.

The method which is responsible for the authorization of the connection is the following:

- **authorizer():**

The method takes in input the packet, to analyze it and determine if it can pass or not.

First of we extract the information we need from the packet. As we can see from the image 4.14, we acquire the IP address and the port both of source and destination. Once we have that information, we will send it as input to the "get\_packet\_details" method, which will deal with the next steps.

```
def drop_or_accept(pkt):
    threshold = 5
    global ip_src, ip_dst, dst_port, src_port
    print(pkt)
    http_packet = scapy.IP(pkt.get_payload())
    #print(http_packet)
    if IP in http_packet:
        ip_src=http_packet[IP].src
        ip_dst=http_packet[IP].dst
        print(ip_dst)
        print(ip_src)
    if TCP in http_packet:
        tcp_sport=http_packet[TCP].sport
        tcp_dport=http_packet[TCP].dport
        print(tcp_dport)
        print(tcp_sport)

    finalTrust = get_packet_details(ip_src, tcp_sport, ip_dst, tcp_dport)
```

**Figure 4.14:** packet information.

At this point, we query the DC to retrieve the information to authorize the communication.

In this case, we want the users authorized to access an IT device. First, we control if the user is authorized to access OT resources. If the user is authorized, we control if he is making the access during working hours (image 4.15).

This is just an example of the controls that can be done to authorize a user, but this simple control can also be very effective. If the access is done outside the employee's working hours, we can assume that it is unauthorized access, and we can block it.

```

if(ipAddr == str(source_address)):
    print("this ip address is correctly associated with", userName)
    if(int(now.hour) >= int(startWork) & int(now.hour) <= int(endWork)):
        print("the work hour is correct")
        deviceTrust += 2
    else:
        deviceTrust -= 2
#deviceTrust = get_device_health(deviceOs, osVersion, deviceTrust)
return deviceTrust

```

**Figure 4.15:** address and work hour control.

We proceed with the authorization. If a user can access the device in the OT, we control if the IP address is the correct one. If the IP is correct, we proceed with a Nmap scan of this IP to retrieve other information.

In our case, we verify the mac address, but we also have other information that could be used to define a context, as we have done in the previous section.

Also in this case, we can control the OS version and if it is affected by some critical vulnerability.

```

now = datetime.datetime.now()
if cur.rowcount == 0:
    print("There is no registered machine with this IP address %s.", source_address)
else:
    for data in cur.fetchone():
        data = cur.fetchone()
        userName = data[0]
        ipAddr = data[12]
        macAddr = data[11]
        startWork = data[2]
        endWork = data[3]
        if(ipAddr == str(source_address)):
            print("the ip address %s is associated with %s", source_address, userName)
            nm.scan(source_address, arguments="-O")
            remoteOs = nm[source_address]['osmatch'][0]['osclass'][0]['osfamily']
            remoteOsVersion = nm[source_address]['osmatch'][0]['osclass'][0]['osgen']
            if(macAddr == remoteOs):

```

**Figure 4.16:** authorizer function part 3.

Finally, if all the controls are done, we will have our final trust level for the connection, and we can authorize or avoid it. This means that we need to manage the packets to leave them to reach or not the PLC in the OT. We can do this using a few lines of code, as shown in Figure 4.17; in this way, depending on the trust level and the threshold, we will give a verdict accepting or dropping the packet.

```
finalTrust = get_packet_details(ip_src, tcp_sport, ip_dst, tcp_dport)
if(finalTrust > threshold):
    print("the final trust level is ", finalTrust)
    print("the connection is allowed")
    pkt.accept()

else:
    pkt.drop()
```

**Figure 4.17:** drop or accept packet.

# Validation

The validation phase is based on testing how the previously described implementation can prevent cyberattacks.

## 5.1 Testbed

To conduct our tests, in particular for the NGFW, we have created a small environment composed of a PLC and a compromised IT machine placed in two different subnets. In this way, we can try to send Modbus commands and validate the ability of the firewall to protect the OT devices.

### 5.1.1 PLC

To simulate the behavior of a PLC, we have used a tool called **PyModSlave**[\[17\]](#), implemented in Python, that can be used to test Modbus slave application for simulation purposes <sup>1</sup>

As said, the tool allows us to test both Modbus RTU and TCP/IP versions, in our case, we have used the second one because it is more recent and secure.

It is straightforward to use; once started, we need to configure the IP address and the port to which we receive commands.

---

<sup>1</sup>PyModSlave starts a TCP/RTU Modbus Slave with 4 data registers: the "coils" and "discrete inputs" registers that are 1-bit data, the first is read and write while the second one is read-only, and the "holding registers" and "input registers" that are 16-bit integers and also, in this case, the first one is a read and write register while the second one is read-only



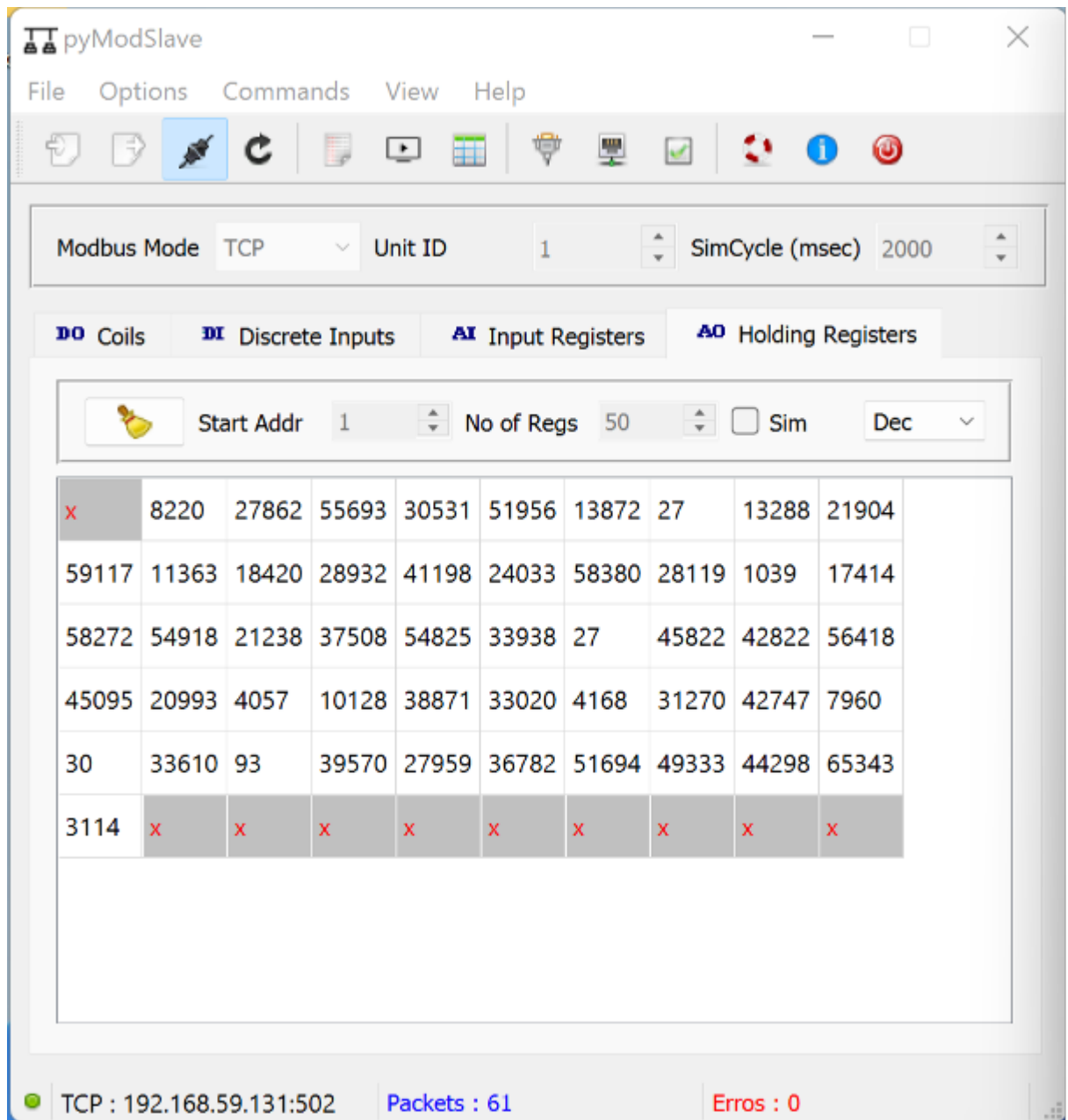


Figure 5.1: PyModSlave.

As we can see from the image 5.1, once configured and started the tool, we have a simple and clear interface that simulates a Modbus slave with the four registers. In this way, we can test writing and reading commands and control if they work correctly.

### 5.1.2 Compromised IT machine

To simulate an IT machine capable of sending Modbus commands, it was necessary to extend the Flask implementation of the Access proxy. After the login, the user

is redirected to a page with two buttons, one to write in a PLC register and one to read from it. To send the Modbus command we have used the library **pymodbus** and in particular the **ModbusTcpClient()** object.

We have defined two functions:

- **writePLC:**

The function in the image 5.2 takes in input the IP address and the port to which send the Modbus command.

```
def writePlc(host_ip, port):
    client = ModbusTcpClient(host_ip, port)
    client.connect()
    data = random.randint(25,35)
    write_reg = random.randint(1,49)
    wr = client.write_registers(write_reg,data,unit=1)
    data = [data , write_reg]
    print(write_reg)
    time.sleep(2)
    return data
```

**Figure 5.2:** writePLC function.

After that, the connection is done successfully, is generated a random value to write in the register, and then send the command with the **write\_register()** method. The command returns a list of “data” that contain at the first position the value that we have written and in the second one the register’s position in which the value was written.

The following images show the request and the response that comes from the slave.

```

> Frame 12: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface bridge100, id 0
> Ethernet II, Src: VMware_bb:04:c2 (00:0c:29:bb:04:c2), Dst: VMware_fe:54:50 (00:0c:29:fe:54:50)
> Internet Protocol Version 4, Src: 192.168.59.130, Dst: 192.168.59.131
> Transmission Control Protocol, Src Port: 45382, Dst Port: 502, Seq: 1, Ack: 1, Len: 15
  Modbus/TCP
    Transaction Identifier: 1
    Protocol Identifier: 0
    Length: 9
    Unit Identifier: 1
  Modbus
    .001 0000 = Function Code: Write Multiple Registers (16)
    Reference Number: 45
    Word Count: 1
    Byte Count: 2
  Register 45 (UINT16): 33
    Register Number: 45
    Register Value (UINT16): 33

```

**Figure 5.3:** Modbus write packet

This showed in the image 5.3 is the write Modbus packet sent to the PLC, in this case, we are writing in the register 45 the number 33.

```

> Frame 13: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface bridge100, id 0
> Ethernet II, Src: VMware_fe:54:50 (00:0c:29:fe:54:50), Dst: VMware_bb:04:c2 (00:0c:29:bb:04:c2)
> Internet Protocol Version 4, Src: 192.168.59.131, Dst: 192.168.59.130
> Transmission Control Protocol, Src Port: 502, Dst Port: 45382, Seq: 1, Ack: 16, Len: 12
  Modbus/TCP
    Transaction Identifier: 1
    Protocol Identifier: 0
    Length: 6
    Unit Identifier: 1
  Modbus
    .001 0000 = Function Code: Write Multiple Registers (16)
    [Request Frame: 12]
    [Time from request: 0.001932000 seconds]
    Reference Number: 45
    Word Count: 1

```

**Figure 5.4:** Modbus write packet response

This in figure 5.4 shows instead the packet that comes from the PLC to confirm that the write command was successful.

- **readPLC:** The other function implemented is "readPLC()" that as we can see from the image 5.5 is very similar to the write function.

```
def readPlc(host_ip, port):
    read_reg= random.randint(1,49)
    client = ModbusTcpClient(host_ip, port)
    connection = client.connect()
    #necessary to working with the proxy
    time.sleep(2)

    rr = client.read_holding_registers(read_reg,1,unit=1)
    data = [json.dumps(rr.registers), read_reg]
    print(data)
    time.sleep(1)
    return data
```

**Figure 5.5:** Modbus readPLC function

An important part of the function is the sleep method after the connection, and this is important to avoid synchronization problems with the proxy's sockets that require some time for the binding process.

In this way, also with this simple implementation, we can avoid that a malicious actor, that has managed to acquire the control of an IT machine, can send commands to the PLC. As said, in this way, we can avoid not only the exfiltration of sensible data from the OT but also that an attacker could send command to a PLC, allowing to cause damages to the production but also putting at risk the safety of the people. Changing also a value in a PLC register, an attacker can modify the behavior, for example, of a mechanical actuator that, based on the type of infrastructure, can be very dangerous.

## 5.2 Validation Access proxy

To test the Access proxy and how the trust algorithm works, we will try three different situations to see how the algorithm work and how contextual information can influence the authentication and authorization phase. In a real scenario, we can set the "trust threshold" dynamically, for example, based on the resources the user wants to access. In particular, we will show how the trust given to the same device can change accordingly with the context.

The first situation is the one in which a user tries to access the resource from a

device that exists in the DB and is trying to log in from a known location.

```
the device is known
The location is correct

Name: Apple macOS Input Validation Error
CVE: CVE-2021-30713
Base score: 7.8
exploitability score: 1.8
impact score: 5.9
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2021-30713
13.7
2

Name: Apple macOS Policy Subsystem Gatekeeper Bypass
CVE: CVE-2021-30657
Base score: 5.5
exploitability score: 1.8
impact score: 3.6
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2021-30657
19.1
3

Name: Apple macOS Out-of-Bounds Write Vulnerability
CVE: CVE-2022-22675
Base score: 7.8
exploitability score: 1.8
impact score: 5.9
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2022-22675
26.8
4

Name: Apple macOS Out-of-Bounds Read Vulnerability
CVE: CVE-2022-22674
Base score: 5.5
exploitability score: 1.8
impact score: 3.6
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2022-22674

The final trust level is 8.05
```

**Figure 5.6:** Device and location known..

We can see in Figure 5.6 that the system has detected that the device and the location are known after it starts to identify the vulnerabilities and if the specific OS version used from the device is affected or not. Each vulnerability has its risk scores, and all those steps influence the trust level of the device that, with this specific context, is "8.05". In this case, the threshold has been set to 8, meaning that the user is authorized and authenticated and can access the resource. The second situation is the one where the device is known, but the location from which it is trying to log in is unknown. As shown in Figure 5.7, the trust of the

device decreases and, in this specific case, the user, even if he has been able to pass the authentication phase, would not be authorized to access the resource due to the low level of trust.

```
the device is known
Wrong location

Name: Apple macOS Input Validation Error
CVE: CVE-2021-30713
Base score: 7.8
exploitability score: 1.8
impact score: 5.9
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2021-30713
10.7
2

Name: Apple macOS Policy Subsystem Gatekeeper Bypass
CVE: CVE-2021-30657
Base score: 5.5
exploitability score: 1.8
impact score: 3.6
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2021-30657
16.1
3

Name: Apple macOS Out-of-Bounds Write Vulnerability
CVE: CVE-2022-22675
Base score: 7.8
exploitability score: 1.8
impact score: 5.9
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2022-22675
23.8
4

Name: Apple macOS Out-of-Bounds Read Vulnerability
CVE: CVE-2022-22674
Base score: 5.5
exploitability score: 1.8
impact score: 3.6
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2022-22674

The final trust level is 7.300000000000001
```

**Figure 5.7:** Device known and location unknown.

Figure 5.8 shows the last situation in which both the device and the location are unknown. As we can see, the information about the device has more weight than the location. This choice was made because the information about the latter is much easier to spoof than the information about the device.

```
unknown device
Wrong location

Name: Apple macOS Input Validation Error
CVE: CVE-2021-30713
Base score: 7.8
exploitability score: 1.8
impact score: 5.9
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2021-30713
1.7000000000000002
2

Name: Apple macOS Policy Subsystem Gatekeeper Bypass
CVE: CVE-2021-30657
Base score: 5.5
exploitability score: 1.8
impact score: 3.6
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2021-30657
7.1000000000000005
3

Name: Apple macOS Out-of-Bounds Write Vulnerability
CVE: CVE-2022-22675
Base score: 7.8
exploitability score: 1.8
impact score: 5.9
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2022-22675
14.8
4

Name: Apple macOS Out-of-Bounds Read Vulnerability
CVE: CVE-2022-22674
Base score: 5.5
exploitability score: 1.8
impact score: 3.6
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2022-22674

The final trust level is 5.0500000000000001
```

**Figure 5.8:** Device and location unknown..

As shown, context-based authentication and authorization can help a lot in the prevention of an attack. In fact, despite all information can be spoofed and everything can be hacked, this solution made the attack much more complex.

### 5.3 Validation NGFW

As also said in the previous chapter, we assume that there is, for maintenance operation, the possibility to send Modbus commands from a machine in the IT to a PLC in the OT. Therefore, the NGFW should understand not only if the user

logged in to the IT machine is authorized to do this type of action but also if the machine from which the connection comes is sufficiently reliable. This is crucial because IT machines are the first to be compromised during an attack on an industrial plant. We will test the ability of the NGFW to block risky connections. We assume that a malicious actor has managed to take control of a machine in the IT, so we are in an advanced step of a typical industrial cyberattack, in which the attacker, through a phishing attack, for example, manages to control a machine inside the company network. If we want to find an analogy with the malware **PIPEDREAM**, we can consider this attack as the PLC Proxy functionality of the **Evilscholar** module, in which the attacker, once that has managed to compromise an Engineering Workstation in the IT (in our case the Engineering workstation is placed in the OT to avoid these situations), he tries to send commands to a PLC in the OT.

With the same procedure, an attacker can use this situation to read/write data from/to the PLCs to exfiltrate information or change values of a PLC registry to create damages. So the idea is that our Zero Trust solution based on a trust algorithm can determine if the connection is authorized or not and block it accordingly. For the first test, we try to send a Modbus packet from a Windows 7 machine, a system no longer supported and with several unpatched vulnerabilities. Once we have sent the command, the NGFW receives the packets and starts the algorithm to determine if the connection can be allowed or not.

First of all, the system determines which employee is logged in to the machine, and then, the DC is queried to determine if the hour of the connection is within the employee's working hours. Once these checks are passed, we verify the device's "health", as shown in Figure 5.9. The mechanism is the same that has been described for the access proxy. In this situation, we can see how the trust score of the windows 7 machine is low, and the connection attempt is rejected from the firewall.



```
this ip address is associated with silviorusso
the work hour is correct

Name: Microsoft Windows Group Policy Privilege Escalation
CVE: CVE-2014-1812
Base score: 9.0
exploitability score: 8.0
impact score: 10.0
attack vector NETWORK
your device version is patched for the vulnerability: CVE-2014-1812

Name: Microsoft Windows Kernel 'Win32k.sys' Local Privilege Escalation Vulnerability
CVE: CVE-2016-0167
Base score: 7.2
exploitability score: 3.9
impact score: 10.0
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2016-0167

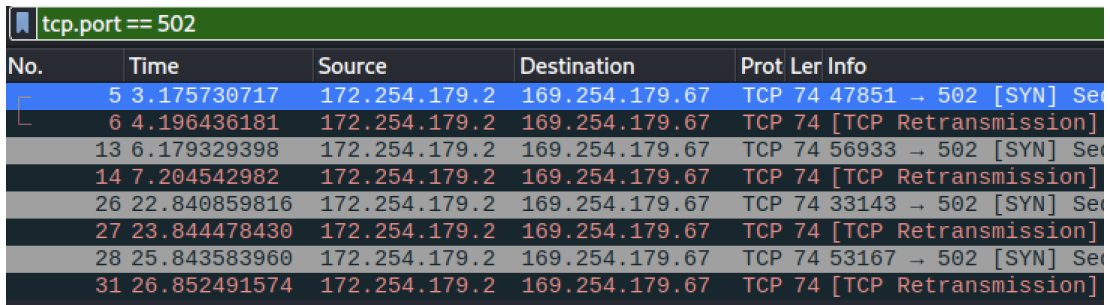
Name: Microsoft Windows Kernel Information Disclosure Vulnerability
CVE: CVE-2021-31955
Base score: 2.1
exploitability score: 3.9
impact score: 2.9
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2021-31955

Name: Microsoft Windows Media Center Remote Code Execution vulnerability
CVE: CVE-2016-0185
Base score: 9.3
exploitability score: 8.6
impact score: 10.0
attack vector NETWORK
your device version is not patched for the vulnerability: CVE-2016-0185

Name: Microsoft Windows Installer Privilege Escalation Vulnerability
CVE: CVE-2020-0683
Base score: 7.2
exploitability score: 3.9
impact score: 10.0
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2020-0683
The final trust level is -9.84
```

**Figure 5.9:** authorization windows machine

In this case, we will issue a negative verdict, and the packet is dropped, blocking any communication to the PLC. This behavior of the firewall can also be seen in Figure 5.10, where after the first TCP packet, the IT machine sends a lot of TCP retransmission, a clear sign that the firewall has done its job.



The image shows a Wireshark packet capture window with a green title bar that reads "tcp.port == 502". The main area displays a list of network packets. The first packet (No. 5) is a SYN packet from 172.254.179.2 to 169.254.179.67 on port 502. Subsequent packets (Nos. 6, 13, 14, 26, 27, 28, 31) are retransmissions of the SYN packet. The status for all these packets is "TCP Retransmission".

No.	Time	Source	Destination	Prot Len Info
5	3.175730717	172.254.179.2	169.254.179.67	TCP 74 47851 → 502 [SYN] Seq
6	4.196436181	172.254.179.2	169.254.179.67	TCP 74 [TCP Retransmission]
13	6.179329398	172.254.179.2	169.254.179.67	TCP 74 56933 → 502 [SYN] Seq
14	7.204542982	172.254.179.2	169.254.179.67	TCP 74 [TCP Retransmission]
26	22.840859816	172.254.179.2	169.254.179.67	TCP 74 33143 → 502 [SYN] Seq
27	23.844478430	172.254.179.2	169.254.179.67	TCP 74 [TCP Retransmission]
28	25.843583960	172.254.179.2	169.254.179.67	TCP 74 53167 → 502 [SYN] Seq
31	26.852491574	172.254.179.2	169.254.179.67	TCP 74 [TCP Retransmission]

**Figure 5.10:** TCP connection blocked from the Firewall.

The second situation is the case of a more updated and secure machine. In this case, shown in Figure 5.11, the trust level of the machine is much higher, and after all the checks, the connection is permitted.

```
this ip address is correctly associated with silviorusso
the work hour is correct

Name: Apple macOS Input Validation Error
CVE: CVE-2021-30713
Base score: 4.6
exploitability score: 3.9
impact score: 6.4
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2021-30713

Name: Apple macOS Policy Subsystem Gatekeeper Bypass
CVE: CVE-2021-30657
Base score: 4.3
exploitability score: 8.6
impact score: 2.9
attack vector NETWORK
your device version is patched for the vulnerability: CVE-2021-30657

Name: Apple macOS Out-of-Bounds Write Vulnerability
CVE: CVE-2022-22675
Base score: 9.3
exploitability score: 8.6
impact score: 10.0
attack vector NETWORK
your device version is patched for the vulnerability: CVE-2022-22675

Name: Apple macOS Out-of-Bounds Read Vulnerability
CVE: CVE-2022-22674
Base score: 4.9
exploitability score: 3.9
impact score: 6.9
attack vector LOCAL
your device version is patched for the vulnerability: CVE-2022-22674
The final trust level is 12.3
the connection is allowed
```

**Figure 5.11:** authorization macOS machine

In this case, we will issue a positive verdict, and the IT machine can communicate with the PLC, as shown in Figure 5.12.

No.	Time	Source	Destination	Protocol
1	0.000000000	172.254.179.2	169.254.179.67	TCP
2	0.020549923	169.254.179.67	172.254.179.2	TCP
3	0.020596789	172.254.179.2	169.254.179.67	TCP
4	0.020873719	172.254.179.2	169.254.179.67	Modbus/TCP
5	0.062766231	169.254.179.67	172.254.179.2	Modbus/TCP
6	0.062807056	172.254.179.2	169.254.179.67	TCP
9	3.070261907	172.254.179.2	169.254.179.67	TCP
10	3.092568711	169.254.179.67	172.254.179.2	TCP
11	3.092707162	169.254.179.67	172.254.179.2	TCP
12	3.092724047	172.254.179.2	169.254.179.67	TCP

Figure 5.12: wireshark: authorized connection

Figure 5.13 shows the PLC holding registers and the packets for the writing operation.

The screenshot shows the pyModSlave software interface. The main window has a menu bar (File, Options, Commands, View, Help) and a toolbar. Below the toolbar, there are settings for 'Modbus Mode' (set to TCP) and 'Unit ID' (set to 1). There are also 'SimCycle (msec)' and 'Dec' buttons. The main area has tabs for 'Coils', 'Discrete Inputs', 'Input Registers', and 'Holding Registers'. The 'Holding Registers' tab is selected, showing a grid of registers. The first register (address 35) has a value of 0. The status bar at the bottom shows 'TCP: 192.168.59.131:502', 'Packets: 12', and 'Errors: 0'. A 'Bus Monitor' window is open on the right, showing 'Raw Data' and 'ADU' details for a received message.

Figure 5.13: pyModSlave: PLC register write

# Conclusions

---

The choice to propose a new Zero Trust solution for the industrial sector comes from a preliminary analysis of literature on this topic that is scarce and mainly focused on the IT sector.

The proposed Zero Trust solution defines a new way to organize an industrial network. We remove any concept of perimeter to rely on a new approach identity-centric, where nothing is taken for known or trusted, and everything must be monitored, authenticated, and authorized. Our solution takes the best from the State-of-the-art of Zero Trust for the IT sector, trying to bring all the benefits of this approach to the OT sector. It can be considered a starting point from which any company can specialize its infrastructure to start the transition to Zero Trust. In the last years, companies have had to choose between connectivity, efficiency, productivity, and security due to all the problems of the industrial sector, and, justifiably, most of the time, the choice falls on the first at the latter's expense. With the state of the art of security for the industrial sector, with I-DMZ and perimeter-based solutions, this is quite impossible to avoid.

This situation is no longer sustainable for the companies, our nations, societies, and our production system, and this is why we have proposed our solution. Zero trust is the only way we can rebalance this situation, allowing companies not to be forced to renounce new technologies to be secure. Talking about the ZTA framework, the fundamental contributions are mainly two, the first one, as discussed, is the elimination of the concept of security perimeter that has lost meaning with today's architectures, the second one, which is perhaps the most important, is the philosophy behind the model.

The Zero Trust model, differently from the other models, does not define any "technical" property. It is not something we can install, instead, it defines a goal to aspire to. Any company can define its own Zero Trust implementation accordingly with its resources and possibility, it is only required that the basic concepts defining the model are respected.

What I could understand is that, differently from the other framework in which the continuous monitoring, the authorization and authentication controls, the risk assessment, and treatment can be considered as "best practices", in zero trust, all

those are embedded in the framework and are indispensable to create a real Zero Trust Architecture.

Suppose we want to define, for example, the privilege level required to access a resource and the Trust algorithm. In that case, we need to define the risk associated with this resource, and to do that, we need to define priorities. A real ZT solution requires a risk assessment to define the company's assets, the associated risk to define priorities, and the level of privilege required to access that resource. We need a fine-grained role assignment, we must monitor and control any access to any resources, define policies and procedures, and everything that is required for a complete cybersecurity plan. The industrial sector can benefit significantly from the Zero Trust model to prevent being overwhelmed by these ever-increasing waves of cyberattacks.

This is, in my opinion, the best benefit which comes from the adoption of a Zero Trust architecture, it is a goal, perhaps unreachable, that push the company to adopt all the best practice required for the protection of their infrastructures accordingly with the maturity level of the specific company.

During the implementation and validation phases, it has become clear that for the transition to a Zero Trust approach, the economic factor is not the main limitation, it does not require an excessive economic effort by companies, and lots of tools are open sources. The real effort is required in time and competencies. All the choices must be correctly planned and examined, and deep knowledge of the infrastructure and all the interactions between the different components is required.

For what concerns future developments, we could integrate our solution with AI technologies. Those would help make the trust algorithm more "smart" and increase its capabilities to understand the context, dynamically adapt to current situations, predict possible attacks, and act more autonomously.

# Bibliography

---

- [1] Shingo Abe et al. “Security threats of Internet-reachable ICS”. In: *2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. 2016, pp. 750–755. DOI: [10.1109/SICE.2016.7749239](https://doi.org/10.1109/SICE.2016.7749239).
- [2] ip-api.org. *ip-api*. URL: <https://ip-api.com> (visited on 2022).
- [3] AT&T. *VLAN Hopping: How to Mitigate an Attack*. URL: <https://cybersecurity.att.com/blogs/security-essentials/vlan-hopping-and-mitigation>.
- [4] Christoph Buck et al. “Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust”. In: *Computers Security* 110 (2021), p. 102436. ISSN: 0167-4048. DOI: <https://doi.org/10.1016/j.cose.2021.102436>. URL: <https://www.sciencedirect.com/science/article/pii/S0167404821002601>.
- [5] CISA. *CISA<sub>A</sub>Alert(AA22 – 103A)*. 2020. URL: <https://www.cisa.gov/uscert/ncas/alerts/aa22-103a>.
- [6] CISA. *Control System Historian*. URL: <https://www.cisa.gov/uscert/ics/Control-System-Historian-Documentation>.
- [7] CISA. *KNOWN EXPLOITED VULNERABILITIES CATALOG*. URL: <https://www.cisa.gov/known-exploited-vulnerabilities-catalog>.
- [8] Cisco/rockwellautomation. *Securely Traversing IACS Data across the Industrial Demilitarized Zone*. 2022. URL: [https://literature.rockwellautomation.com/idc/groups/literature/documents/td/enet-td009\\_en-p.pdf](https://literature.rockwellautomation.com/idc/groups/literature/documents/td/enet-td009_en-p.pdf).
- [9] Claroty. *80% of Critical Infrastructure Organizations Experienced Ransomware Attacks Last Year*. URL: <https://claroty.com/press-releases/80-of-critical-infrastructure-organizations-experienced-ransomware-attacks-last-year>.
- [10] CrowdStrike. *A frictionless Zero Trust approach for the enterprise*. URL: <https://www.crowdstrike.com/resources/data-sheets/frictionless-zero-trust-approach-for-enterprises-solution-brief>.
- [11] C. Cunningham. “The Zero Trust eXtended (ZTX) ecosystem”. In: (2018).

- [12] Casimer DeCusatis et al. “Implementing Zero Trust Cloud Networks with Transport Access Control and First Packet Authentication”. In: *2016 IEEE International Conference on Smart Cloud (SmartCloud)*. 2016, pp. 5–10. DOI: [10.1109/SmartCloud.2016.22](https://doi.org/10.1109/SmartCloud.2016.22).
- [13] GE digital. *Industrial Digital Twins: Real Products Driving \$1B in Loss Avoidance*. 2020. URL: <https://www.ge.com/digital/blog/industrial-digital-twins-real-products-driving-1b-loss-avoidance>.
- [14] Shalanda D. Young (Acting Director). *HOMELAND SECURITY PRESIDENTIAL DIRECTIVE 7: CRITICAL INFRASTRUCTURE IDENTIFICATION, PRIORITIZATION, AND PROTECTION*. 2003. URL: <https://www.cisa.gov/homeland-security-presidential-directive-7>.
- [15] Shalanda D. Young (Acting Director). *Moving the U.S. Government Toward Zero Trust Cybersecurity Principles*. 2022. URL: <https://www.whitehouse.gov/wp-content/uploads/2022/01/M-22-09.pdf>.
- [16] Dragos. *Pipedream*. 2022. URL: <https://www.dragos.com/blog/industry-news/chernovite-pipedream-malware-targeting-industrial-control-systems> (visited on 04/13/2022).
- [17] ElBar. *PyModSlave*. URL: <https://pypi.org/project/pyModSlave/>.
- [18] Herve Eychenne. *iptables-Linux man page*. URL: <https://linux.die.net/man/8/iptables> (visited on 2021).
- [19] Wengao Fang and Xiaojuan Guan. “Research on iOS Remote Security Access Technology Based on Zero Trust”. In: *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)*. Vol. 6. 2022, pp. 238–241. DOI: [10.1109/ITOEC53115.2022.9734455](https://doi.org/10.1109/ITOEC53115.2022.9734455).
- [20] Luca Ferretti et al. “Survivable zero trust for cloud computing environments”. In: *Computers Security* 110 (2021), p. 102419. ISSN: 0167-4048. DOI: <https://doi.org/10.1016/j.cose.2021.102419>. URL: <https://www.sciencedirect.com/science/article/pii/S0167404821002431>.
- [21] Flask. *Flask project*. 2022. URL: <https://flask.palletsprojects.com/en/2.2.x> (visited on 2022).
- [22] Mark Goudie. *Going Beyond Malware, The Rise of “Living off the Land” Attacks*. URL: <https://www.crowdstrike.com/blog/going-beyond-malware-the-rise-of-living-off-the-land-attacks/>.



- [23] Andy Greenberg. *A Hacker Tried to Poison a Florida City's Water Supply*. 2021. URL: <https://www.wired.com/story/oldsmar-florida-water-utility-hack/>.
- [24] U.S. Department of Homeland security. *Recommended Practice: Improving Industrial Control System Cybersecurity with Defense-in-Depth Strategies*. 2016. URL: [https://www.cisa.gov/uscert/sites/default/files/recommended\\_practices/NCCIC\\_ICS-CERT\\_Defense\\_in\\_Depth\\_2016\\_S508C.pdf](https://www.cisa.gov/uscert/sites/default/files/recommended_practices/NCCIC_ICS-CERT_Defense_in_Depth_2016_S508C.pdf).
- [25] *IEC/TS 62443 Industrial communication networks – Network and system security – Part 1-1: Terminology, concepts and models*. Standard. INTERNATIONAL ELECTROTECHNICAL COMMISSION.
- [26] “IEEE Standard for Local and metropolitan area networks–Bridges and Bridged Networks–Amendment 30: YANG Data Model”. In: *IEEE Std 802.1Qcp-2018 (Amendment to IEEE Std 802.1Q-2018)* (2018), pp. 1–93. DOI: [10.1109/IEEESTD.2018.8467507](https://doi.org/10.1109/IEEESTD.2018.8467507).
- [27] Federal Bureau of Investigation (FBI). *TRITON Malware Remains Threat to Global Critical Infrastructure Industrial Control Systems (ICS)*. 24-03-2022. URL: <https://www.ic3.gov/Media/News/2022/220325.pdf>.
- [28] kaspersky. *Industroyer v1*. URL: <https://www.kaspersky.com/enterprise-security/mitre/industroyer>.
- [29] Patrick Katuruza. *IT OT Convergence, Managing the Cybersecurity Risks*. URL: <https://gca.isa.org/blog/it-ot-convergence-managing-the-cybersecurity-risks>.
- [30] Sreejith Keeriyattil. “NSX Service Composer and Third-Party Integration”. In: *Zero Trust Networks with VMware NSX: Build Highly Secure Network Architectures for Your Data Centers*. Berkeley, CA: Apress, 2019. ISBN: 978-1-4842-5431-8. DOI: [10.1007/978-1-4842-5431-8\\_4](https://doi.org/10.1007/978-1-4842-5431-8_4). URL: [https://doi.org/10.1007/978-1-4842-5431-8\\_4](https://doi.org/10.1007/978-1-4842-5431-8_4).
- [31] Qin Y. Khan S. Parkinson S. *Fog computing security, a review of current applications and security solutions*. 2017. URL: <https://doi.org/10.1186/s13677-017-0090-3>.
- [32] Brian Knowlton. *Honeywell Warns of Increasing Attacks by State-Sponsored Hackers*. URL: <https://www.bloomberg.com/news/articles/2016-06-03/honeywell-warns-of-increasing-attacks-by-state-sponsored-hackers#xj4y7vzkg>.

- [33] Brian Knowlton. *Military Computer Attack Confirmed*. URL: <https://www.nytimes.com/2010/08/26/technology/26cyber.html>.
- [34] Ralph Langner. “Stuxnet: Dissecting a Cyberwarfare Weapon”. In: *IEEE Security Privacy* 9.3 (2011), pp. 49–51. DOI: [10.1109/MSP.2011.67](https://doi.org/10.1109/MSP.2011.67).
- [35] Federico Magnanini, Luca Ferretti, and Michele Colajanni. “Flexible and Survivable Single Sign-On”. In: Jan. 2022, pp. 182–197. ISBN: 978-3-030-94028-7. DOI: [10.1007/978-3-030-94029-4\\_13](https://doi.org/10.1007/978-3-030-94029-4_13).
- [36] Federico Magnanini, Luca Ferretti, and Michele Colajanni. “Scalable, Confidential and Survivable Software Updates”. In: *IEEE Transactions on Parallel and Distributed Systems* 33.1 (2022), pp. 176–191. DOI: [10.1109/TPDS.2021.3090330](https://doi.org/10.1109/TPDS.2021.3090330).
- [37] Mandiant. *INCONTROLLER: New State-Sponsored Cyber Attack Tools Target Multiple Industrial Control Systems*. 13-04-2022. URL: <https://www.mandiant.com/resources/blog/incontroller-state-sponsored-ics-tool>.
- [38] Mandiant. *INDUSTROYER.V2: Old Malware Learns New Tricks*. 25-03-2022. URL: <https://www.mandiant.com/resources/blog/industroyer-v2-old-malware-new-tricks>.
- [39] MIT Martin Giles. *Triton is the world’s most murderous malware, and it’s spreading*. 5-03-2019. URL: <https://www.technologyreview.com/2019/03/05/103328/cybersecurity-critical-infrastructure-triton-malware>.
- [40] Microsoft. *Application proxy, how it work*. 2022. URL: <https://docs.microsoft.com/en-us/azure/active-directory/app-proxy/what-is-application-proxy#application-proxy-connectors>.
- [41] Microsoft. *Azure AD application proxy*. 2022. URL: <https://docs.microsoft.com/en-us/azure/active-directory/app-proxy/application-proxy>.
- [42] Microsoft. *How an IoT Edge device can be used as a gateway*. 2020. URL: <https://docs.microsoft.com/en-us/azure/iot-edge/iot-edge-as-gateway?view=iotedge-2020-11>.
- [43] Microsoft. *IoT concepts and Azure IoT Hub*. 2022. URL: <https://docs.microsoft.com/en-us/azure/iot-hub/iot-concepts-and-iot-hub>.

- [44] Microsoft. *Iot edge as gateway*. 2020. URL: <https://docs.microsoft.com/en-us/azure/iot-edge/iot-edge-as-gateway?view=iotedge-2020-11>.
- [45] Microsoft. *Microsoft Defender for Cloud*. 2022. URL: <https://docs.microsoft.com/en-us/azure/defender-for-cloud/defender-for-cloud-introduction>.
- [46] Microsoft. *Microsoft Defender for IoT*. 2022. URL: <https://azure.microsoft.com/en-us/services/iot-defender/#overview>.
- [47] Microsoft. *Microsoft threat intelligence*. 2022. URL: <https://www.microsoft.com/en-us/insidetrack/microsoft-uses-threat-intelligence-to-protect-detect-and-respond-to-threats>.
- [48] Microsoft. *Using the Organizational Domain Forest Model*. 2021. URL: <https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/using-the-organizational-domain-forest-model#domain-owner> (visited on 07/29/2021).
- [49] Microsoft. *What is Azure IoT Edge*. 2020. URL: <https://docs.microsoft.com/en-us/azure/iot-edge/about-iot-edge?view=iotedge-2020-11>.
- [50] MITRE. *CPE*. URL: <https://cpe.mitre.org/about/>.
- [51] MITRE. *CVE® Program Mission*. URL: <https://www.cve.org>.
- [52] netfilter.org. *Iptables*. URL: <https://www.netfilter.org> (visited on 2021).
- [53] NIAC. *CVSS*. URL: <https://nvd.nist.gov/vuln-metrics/cvss>.
- [54] NIST. *NATIONAL VULNERABILITY DATABASE*. URL: <https://nvd.nist.gov/developers/vulnerabilities>.
- [55] NIST. *Zero Trust Architecture*. 2020. URL: <https://doi.org/10.6028/NIST.SP.800-207> (visited on 10/2020).
- [56] Nmap. *Nmap project*. 2022. URL: <https://nmap.org/> (visited on 2022).
- [57] Nmap.org. *Legal Issues*. 2020. URL: <https://nmap.org/book/legal-issues.html>.
- [58] Nmap.org. *Remote OS Detection*. URL: <https://nmap.org/book/osdetect.html#idm45323735721296> (visited on 2022).
- [59] OpenVPN. *OpenVPN*. URL: <https://openvpn.net>.

- [60] Paloalto. *Applying VLAN Insertion in ICS/SCADA*. 2020. URL: <https://www.paloaltonetworks.com/resources/whitepapers/applying-vlan-insertion-in-ics-scada> (visited on 07/29/2020).
- [61] Paloalto. *PA-5200 Series datasheet*. 2022. URL: <https://www.paloaltonetworks.com/resources/datasheets/pa-5200-series-specsheet> (visited on 03/23/2022).
- [62] Paloalto. *What Is Zero Trust for the Cloud?* URL: <https://www.paloaltonetworks.com/cyberpedia/what-is-a-zero-trust-for-the-cloud>.
- [63] Python. *Python urllib-request*. URL: <https://docs.python.org/3/library/urllib.request.html>.
- [64] V. Radha and D. Hitha Reddy. "A Survey on Single Sign-On Techniques". In: *Procedia Technology* 4 (2012). 2nd International Conference on Computer, Communication, Control and Information Technology( C3IT-2012) on February 25 - 26, 2012, pp. 134–139. ISSN: 2212-0173. DOI: <https://doi.org/10.1016/j.protcy.2012.05.019>. URL: <https://www.sciencedirect.com/science/article/pii/S2212017312002988>.
- [65] RADIUS. *RADIUS*. URL: <https://en.wikipedia.org/wiki/RADIUS>.
- [66] ReFirmLabs. *Binwalk*. 2022. URL: <https://github.com/ReFirmLabs/binwalk>.
- [67] SANS. *The Industrial Control System Cyber Kill Chain*. 2020. URL: <https://www.sans.org/white-papers/36297/>.
- [68] Siemens. *TIA portal*. URL: [https://new.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html?acz=1&gclid=CjwKCAjw6raYBhB7EiwABge5Kir08VtJWWZU803\\_CgBNI0YZG6T1tQ2ugOSPjJKLOW3xH6LidMSrUhoCHgAQAvD\\_BwE](https://new.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html?acz=1&gclid=CjwKCAjw6raYBhB7EiwABge5Kir08VtJWWZU803_CgBNI0YZG6T1tQ2ugOSPjJKLOW3xH6LidMSrUhoCHgAQAvD_BwE)).
- [69] stong. *CVE-2020-15368*. 2020. URL: <https://www.sans.org/white-papers/36297/>.
- [70] VMware. *VMware*. URL: <https://www.vmware.com/>.
- [71] VMware. *What is Zero Trust Edge?* URL: <https://www.vmware.com/topics/glossary/content/zero-trust-edge.html>.
- [72] Wikipedia. *OSI Model*. URL: [https://en.wikipedia.org/wiki/OSI\\_model](https://en.wikipedia.org/wiki/OSI_model).
- [73] Wikipedia. *Sheep dip*. URL: [https://en.wikipedia.org/wiki/Sheep\\_dip\\_\(computing\)](https://en.wikipedia.org/wiki/Sheep_dip_(computing)).

- 
- [74] T.J. Williams. “The Purdue Enterprise Reference Architecture”. In: *IFAC Proceedings Volumes* 26.2, Part 4 (1993). 12th Triennial World Congress of the International Federation of Automatic control. Volume 4 Applications II, Sydney, Australia, 18-23 July, pp. 559–564. ISSN: 1474-6670. DOI: [https://doi.org/10.1016/S1474-6670\(17\)48532-6](https://doi.org/10.1016/S1474-6670(17)48532-6). URL: <https://www.sciencedirect.com/science/article/pii/S1474667017485326>.
- [75] Linjiang Xie et al. “A Micro-Segmentation Protection Scheme Based on Zero Trust Architecture”. In: *ISCTT 2021; 6th International Conference on Information Science, Computer Technology and Transportation*. 2021, pp. 1–4.
- [76] Zscaler. *VPN Risk Report*. 2021. URL: <https://info.zscaler.com/rs/306-ZEJ-256/images/2021-VPN-Risk-Report-Zscaler.pdf>.