2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA) | 979-8-3503-2385-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/TPS-ISA58951.2023.00034

Model Based Risk Assessment and Risk Mitigation Framework for Cyber-Physical Systems

1st Shwetha Gowdanakatte
Department of Systems Engineering
Colorado State University
Fort Collins, U.S.A.
Shwetha.Gowdanakatte@colostate.edu

2nd Indrakshi Ray
Department of Computer Science
Colorado State University
Fort Collins, U.S.A.
Indrakshi.Ray@colostate.edu

3rd Mahmoud Abdelgawad
Department of Computer Science
Colorado State University
Fort Collins, U.S.A.
m.abdelgawad@colostate.edu

***Abstract*—Cyber-Physical Systems (CPS) form the nation's critical infrastructure. The present-day CPS incorporate advanced communication technologies for remote monitoring and control. However, CPS often have infrastructures that have a very long lifespan. Consequently, the incorporation of cybersecurity in legacy systems is a challenge. This has led to an increased attack surface for CPS. Identifying risks and implementing security techniques during the design phase can help improve the security posture of CPS. This paper presents a holistic model-based risk assessment and mitigation framework for CPS to achieve security by design. Such a framework can also be applied to legacy systems to mitigate the risks.**

***Index Terms*—Cybersecurity, Cyber-Physical Systems, Unified Modeling Language, System Dynamics, Causal Loop Diagrams, Zero Trust Architecture.**

I. Introduction

Cyber Physical Systems (CPS) form a nation's critical infrastructure, such as electrical power grids, oil and natural gas distribution and transportation systems, and healthcare devices. With the rapid advancement of computing and communication infrastructures, Operational Technology (OT) systems in CPS are now connected with Information Technology (IT) systems. Such connectivity allows for remote monitoring and controlling entities in the physical environment.

Integrating OT with IT systems have increased the attack surface in CPS. This is evidenced by the fact that cyberattacks on CPS have significantly increased in recent years, particularly on power grids. Stuxnet attack was one of the complex cyberattacks that targeted the Iranian power grid in 2010 [1]. Other examples include the attack on the Ukrainian electrical substations in April 2022 [2], the attack on German wind farms in 2022 [3], the Red Echo Advanced Persistent Threat (APT) on Indian power sectors in 2021 [4], the attack on the European Network of Transmission System Operators for Electricity (ENTSO-E) in 2020, the Russian power grid in 2019, and the Saudi Armaco petrochemicals in 2017 [1].

The root cause of increasing cyberattacks on CPS is the failure to identify and address the cyber risks in the design phase. Identifying risks in given CPS and addressing them goes beyond the individual component. The CPS are complex systems with interconnected IT, OT, and physical assets. Cyberattack on a single asset may not pose a threat to it, but the cascading effect of the attack may be catastrophic. A highly skilled or motivated attacker may use a multi-vector attack that exploits the vulnerability of an asset that can lead to chain effects of the attack on the entire targeted CPS. Many attacks have evidenced this. For instance, the cyberattack launched on satellite communication caused the malfunction of wind turbines in German wind farms [3]. The attack on an unpatched server eventually caused a Denial of Service (DoS) attack by disrupting the communication between the main control center and remote power generation sites [5].

Our preliminary investigation finds that research to date focuses on specific aspects of the cybersecurity of CPS. We need to provide a holistic view of the cybersecurity framework. There are insufficient methods to identify the interconnection between the critical assets, component-level vulnerabilities, possible attacks on critical assets, and their cascading effects. Additionally, the risk management methodologies adapted by the organizations are manual, which is tedious, error-prone, and time-consuming. There is no formalized unified framework to automatically identify the critical assets, their exploitable vulnerabilities, attacks, and the cascading effects of the attacks.

This paper presents a novel holistic unified risk assessment and mitigation framework that bridges the gaps in the current research. The framework incorporates (i) an automatic model-based risk assessment and (ii) a modular zero trust risk mitigation technique.

Model-Based Design (MBD) has been proven to be an efficient technique for CPS to understand the physical and logical relationship between the various assets. MBD helps with security analysis in the design phase; it helps in separating security concerns, traceability, impact analysis, formal verification, and simulation [6]. Towards this end, our automated risk assessment involves modeling the given CPS using the Unified Modeling Language (UML). The UML provides a visual framework of CPS components, attributes, and interconnections. It also facilitates programmatic analysis of the given CPS architecture.

Using the UML as a basis, our risk assessment method develops algorithms for the automatic identification of critical assets and automatic vulnerability assessment. It incorporates Causal Loop Diagrams (CLD) to visually represent possible threats' chain effects on the entire CPS.

979-8-3503-2385-6/23/$31.00 ©2023 IEEE
DOI 10.1109/TPS-ISA58951.2023.00034

Our framework helps organizations identify the risks and possible threats in the design phase to implement proper security controls to avoid the catastrophic effects of cyberattacks.

The risk mitigation incorporates a zero-trust architecture which addresses most of the vulnerabilities identified during the risk assessment. Our zero-trust architecture is designed on the principle of "never trust, always verify." It verifies every request for authentication, deep packet inspection, matching patterns with already known attacks, and access control. The traditional Zero Trust model is challenging to implement for complex event-driven CPS. We present a Zero Trust Architecture (ZTA) that can be integrated as an independent module without modifying the CPS architecture.

The rest of the paper is organized as follows. We proceed with exploring the findings of our studies on current research in Section II . We then provide an overview of our proposed model for risk assessment and mitigation techniques in Section III. We discuss the details of risk assessment framework in Section IV. We then discuss the architecture of ZTA as plug and play module and provide a detailed security analysis in Section V. We conclude with future enhancements for the proposed methodology in Section VI.

II. Related Work

The National Institute of Standards and Technology (NIST) provides a Risk Management Framework (RMF) to assess and integrate security policies into the system development life cycle [7]. In addition to cybersecurity research organizations, academic researchers have made significant efforts to address the cybersecurity problems of CPS by proposing various methodologies for assessing risks and providing mitigation techniques.

Akbarzadeh et al. [8] propose a domain-agnostic, dependency-based risk assessment framework for large-scale CPS. The proposed methodology models the system using graph theory; it then identifies the critical components based on Closeness Centrality, Tacit Input Centrality, and Tacit Output Centrality [9] and performs risk assessment on the identified critical assets. This paper focuses on conducting the risk assessment for individual critical assets without considering the entire system.

Young et al. [10] propose a manual methodology for assessing the risks of a CPS under consideration. The proposed methodology defines the different layers of the CPS under consideration, identifies the critical assets based on domain knowledge, identifies vulnerabilities, and quantifies the risk based on subjective scores.

Kawanishi et al. [11], Semertzis et al. [12], and Zhang et al. [13] propose methods and mathematical modeling for quantitative risk assessment of CPS. These methods are based on the subjective score and the domain-specific information. They incorporate manual scenario-based qualitative risk analysis to calculate the subjective score.

Sundararajan et al. [14] conduct a comprehensive review of vulnerabilities and attacks at the protocol level for solar and wind systems and propose solutions to improve protocol-level security. This paper mainly focuses on the IT protocols at the OSI network reference model. Krause et al. [15] focus on the OT security of CPS. This paper presents a security analysis of the OT protocols commonly used in CPS network infrastructure. It provides comprehensive attack vectors that measure the identified attacks' scope, severity, difficulty, and impact. Zografopoulos et al. [16] conduct a detailed study of CPS that includes various subsystems. This paper presents a novel threat model incorporating adversary and attack models. The adversary model helps identify adversarial knowledge, resources, access, and specificity. The attack model helps identify critical assets, techniques, and tactics for possible attacks.

Zhang et al. [17] provides a quantitative analysis of Mean Time To Compromise (MTTC) taken by a possible attack which is unique as most works do not address the temporal component of cyberattacks.

Gunduz et al. [18] present the mental model of attackers, which is not presented in the rest of the papers. The mental model of attackers is an essential aspect of cybersecurity. Attackers view their targeted system holistically to exploit the vulnerabilities and find the weak links within a system. Understanding their mental model, motivation, and skill sets helps improve cybersecurity for any critical infrastructure.

Gowdanakatte et al. [19] present an Attribute-Based Access Control (ABAC) gateway for addressing the access control vulnerabilities in Programmable Logic Controllers (PLC). Although this paper presents a robust security model based on NGAC-ABAC, it prevents only access control-related threats. We extend the work of Gowdanakatte et al. [19] to implement a holistic Zero Trust Architecture (ZTA) for the other assets in the given CPS, including the PLC. We incorporate additional modules to the ABAC gateway to address access control-related vulnerabilities and other vulnerabilities.

III. Proposed Model

Our risk management framework consists of risk assessment and risk mitigation phases described below.

A. Phase 1: Automated Risk Assessment

[Process 1: Reconnaissance and UML modeling of the CPS network architecture] We start our assessment with a detailed study of the network architecture of the CPS under consideration. This process focuses on modeling the network architecture with Unified Modeling Language (UML). UML visually represents assets, attributes, functions, and interconnections. It also facilitates programmatic extraction of information required for the risk assessment process [20].

[Process 2: Identification of critical assets] The critical assets are the assets that are essential to perform the system function without any interruption, and their failure is not tolerated. Distinguishing critical assets in CPS is vital and helps organizations to develop a risk mitigation plan to avoid catastrophic consequences. Manual identification of critical assets is tedious and error-prone. Hence, this process focuses on

the semi-automatic identification and prioritization of critical assets. We construct a network graph from the UML model and develop an algorithm that determines the assets' criticality based on the network graph. We discuss the algorithm in Section IV.

[Process 3: Identification and automatic assessment of vulnerabilities] This process identifies the vulnerabilities in each critical asset. We use a library implemented in Python to extract the vulnerabilities from the NIST vulnerability database. The Python scripts extract all the vulnerabilities and their corresponding details from the database. There may be multiple vulnerabilities for a given critical asset, and they may not be exploitable. In order to identify the exploitable vulnerabilities, we derive the network topology from the UML modeling and design algorithms to automatically identify the exploitable vulnerabilities and generate the attack graphs. We discuss the detailed algorithm for automatic vulnerability assessment in Section IV.

[Process 4: Modeling the cascading effect of potential attacks] A cyber attack on a critical asset can have a cascading impact on other assets due to its physical or logical connections. It may lead to multiple component failures, eventually causing the failure of the entire given CPS. Understanding the cascading effect of an attack on a given critical asset is vital to place proper security techniques to avoid the catastrophic failure of the system. Therefore, the next process is to understand the cascading effect of an attack on a given critical asset. We use Causal Loop Diagrams (CLD) to analyze an attack's cascading effect. Causal Loop Diagrams are System Dynamics tools. They are used to understand the influence of interconnected components on each other. [21]. We generate the CLD from the attack graphs and critical assets obtained in the previous steps. A detailed CLD is discussed in Section IV.

B. Phase 2: Risk Mitigation with Zero Trust Architecture

In this phase, we integrate Zero Trust Architecture (ZTA) with the given CPS to address the potential attacks exploiting the identified vulnerabilities. ZTA provides a collection of principles and policies that enforce the least privilege per request access in IT and OT components. NIST provides guidance for implementing ZTA. The traditional ZTA typically consists of a policy engine (PE), policy administrator (PA), and policy enforcement point (PEP) components to enforce access policy per access. Additionally, it can have multiple layers of security, such as network segmentation and deep packet inspection. The existing ZTA architecture is focused on IT security and is complex to implement for event-driven CPS. Additionally, it does not consider OT protocols and the component-level vulnerabilities of OT devices.

Our method provides a novel plug-and-play ZTA that fits best for the CPS environment. We implement the ZTA as an independent gateway between the critical asset and its access point. It provides protection against unauthorized access to the assets through Attribute Based Access Control. Consequently, many common attacks such as unintended modifications and other malicious operations through deep packet inspection, repudiation attacks, denial of service attacks, integrity attacks, and elevation of privileges do not occur.

IV. Phase 1: Risk Assessment

We consider a wind farm system as a reference CPS architecture for developing our risk assessment model.

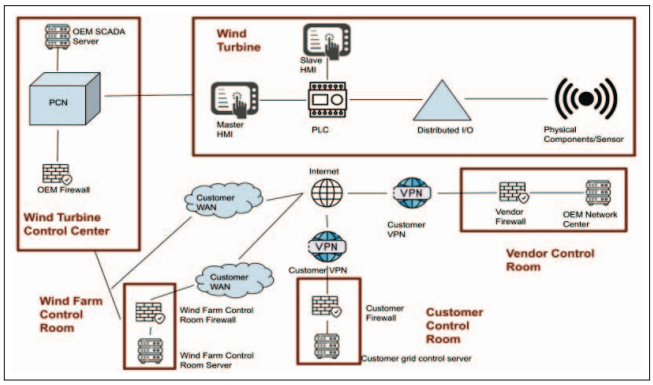

Fig. 1: Network Topology in a Wind Farm

A. Process 1: Reconnaissance and UML Modeling of the CPS Network Architecture

A typical wind farm network topology is shown in Figure 1. It comprises different subsystems denoted by the rectangular blocks outlined in red color.

Wind Turbine subsystem consists of a Programmable Logic Controller (PLC) that communicates with a slave Human Machine Interface (HMI), master HMI, and distributed inputs/outputs through an Ethernet switch. Distributed inputs/outputs communicate with physical sensors through physical connection for sending or receiving operation-specific information.

Wind Turbine Control Center subsystem consists of the Original Equipment Manufacturer (OEM) Supervisory Control And Data Acquisition (SCADA) server that communicates with the wind turbine through Process Control Network (PCN) using industrial protocols such as MODBUS, PROFINET, and DNP3.

Wind Farm Control Room subsystem has servers communicating with wind turbines using SCADA protocols on the Demilitarized Zone (DMZ).

Customer Control Room subsystem allows the remote customer access to the wind turbine through a local control center via a secured Virtual Private Network (VPN) connection.

Vendor Control Room subsystem allows the remote vendor access to the wind turbines for remote troubleshooting and product updates through a secure VPN.

The reference architecture uses 1) Siemens S71500 PLC and Siemens HMI for control devices. 2) Windows Server 2012 for the OEM SCADA Server, the wind farm control room server, and the customer grid control server. 3) PROFINET industrial

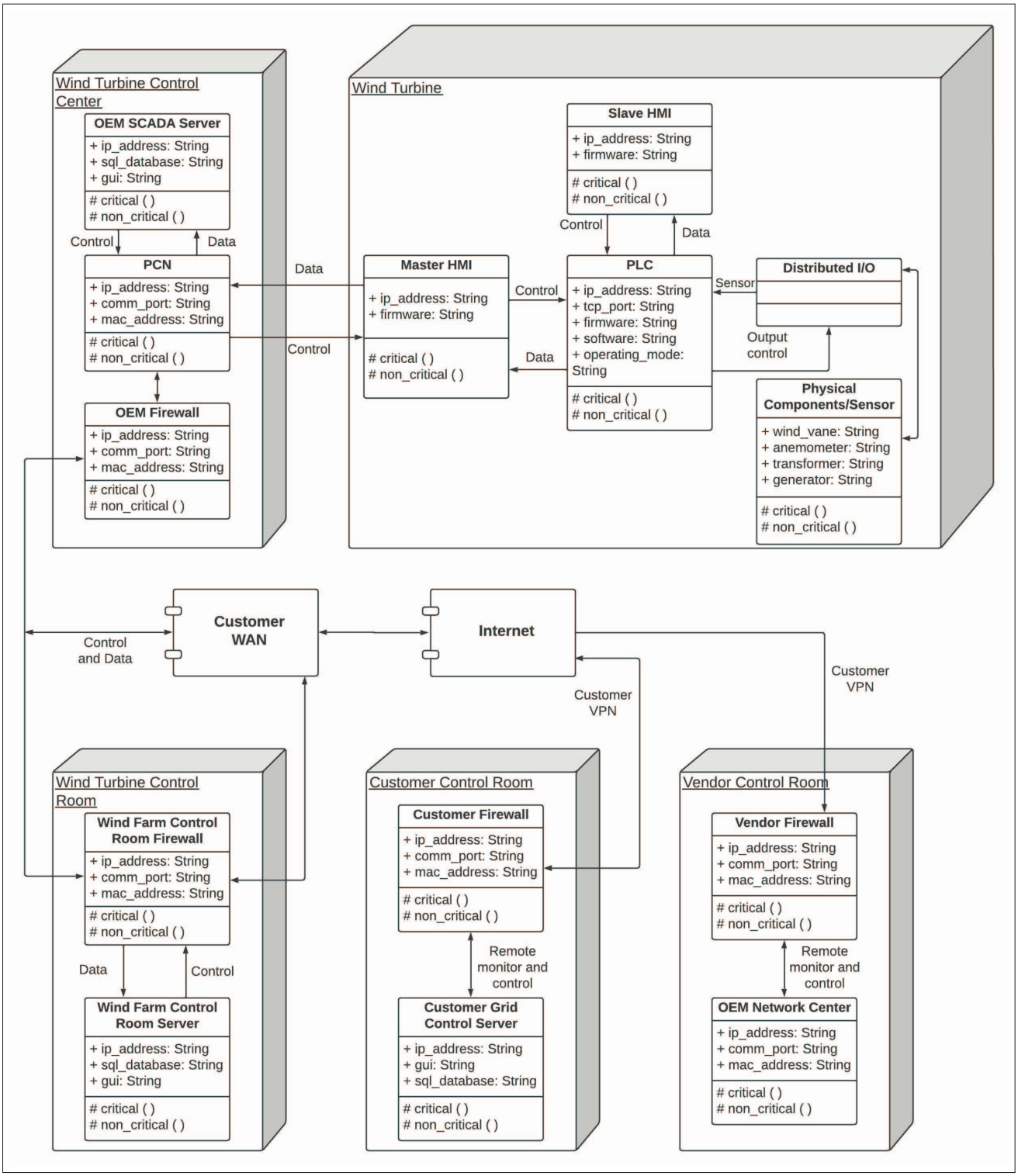

Fig. 2: UML Diagram for Wind farm architecture

protocol for Process Control Network (PCN). 4) Cisco VPNs and firewalls.

The wind farm architecture shown in Figure 1 is a static topology that only visually represents different components

and their interconnections. We cannot programmatically extract the required information about assets and their interconnections for the automatic risk assessment. Hence, we develop a UML modeling of the given network architecture. Using UML, we can programmatically extract information about the system's components, features, functions, and interconnections [20]. We use a Python library Python to Plant UML [22] to generate the UML model of the wind farm network architecture.

Figure 2 represents the UML modeling of a wind farm architecture. The outer rectangular boxes denote subsystems of the wind farm. The subsystems shown in the figure are (i) Wind turbine. (ii) Wind turbine control center. (iii) Wind farm control room. (iv) Vendor control room. (v) Customer control room. The rectangular blocks within a subsystem denote the assets' class. The first partition of the rectangular block denotes an asset, which we refer to as *Asset*. The second partition denotes the attribute list. The attributes are prefixed with $+$. We refer to the attribute as *Attr*. The last partition denotes the functions. The functions are prefixed with #. We refer to functions as *Func*. The edges connecting the assets denote the connections between them. We refer to edges as *Edge*. For example, the wind turbine subsystem consists of the following assets with their attributes and functions.

PLC Attributes are (i) $+$*type* denotes the asset type. The asset type of the PLC is *ControlDevice*. (ii) $+$*ip_address* denotes the PLC's IP address. (iii) $+$*tcp_port*= denotes the communication port. (iv) $+$*firmware*= represents the firmware version. (v) $+$*software*= represents software description. (vi) $+$*Operating_mode*= {*Stopped, Running,Remote* } represents current operational status of the PLC. *# critical()*, and *# non_critical()* are the functions in PLC software.

Master HMI The attributes are (i) $+$*type* is of type string that denotes the asset type. The asset type of the HMI is *ControlDevice* (ii) $+$*ip_address* (iii) $+$*firmware* denotes HMI firmware version. (iv) *# critical()*, and *# no_critical()* are the functions in the HMI software.

Physical Components The attributes are (i) $+$*type* is of type string that denotes the asset type. The asset type of the Physical Component is *PhysicalDevice* (ii) $+$ *name*= {*transformer, generator,anemometer* } is of type string that denotes the asset name. (iii) *# critical()*, and *# no_critical()* are the functions.

The *Edge. Control* and *Edge. Data* denote the interconnections between *PLC*, and *Master HMI*, or *Slave HMI*. The HMIs communicate control parameters to the *PLC* through *Edge. Control*. The *PLC* communicates the operational data to the HMIs through *Edge. Data*.

B. Process 2: Identification of Critical Assets

A critical asset is an asset that is essential to perform the system's function without interruption. Compromising critical assets can lead to loss of CPS's availability, integrity, and confidentiality and cause serious consequences. We use domain knowledge for manual and graph theory to automatically identify critical assets.

1) Manual Identification

The wind farm architecture has the following critical assets.

SCADA Servers: SCADA servers contain critical data from wind farm operations, such as the previous history of alarms, events, and data related to power generation and transmission. The attack on SCADA servers can lead to loss of integrity, leading to the malfunction of the wind turbines.

Virtual Private Network (VPN): VPNs provide secure communication between wind farms and remote control centers. Attacks on VPN servers can provide unauthorized access to the wind farm network. Attackers can send crafted messages to the wind farm through compromised VPN servers to cause Denial Of Service (DoS) attacks.

Control Devices (PLC, HMIs, Servo Drives, VFDs): Control devices in each wind turbine are critical assets as they monitor and control wind turbine operations. Cyberattacks on these devices have a direct impact on critical operations. Examples of impact include inverter shutdowns, voltage sags, and uncontrolled pitch.

Communication Protocols: The communication protocols are intangible cyber assets. Control devices use vendor-specific protocols wrapped in TCP/IP packets to communicate with other devices. Attackers can exploit the vulnerabilities in the vendor-specific protocols to send crafted messages to wind turbines to cause Availability, Integrity, or Confidentiality (AIC) attacks. Thus, it is required to assess the vulnerabilities in the communication protocols at both the IT and OT level for better security posture. Examples of OT protocols are S7P3 (a proprietary application layer protocol by Siemens) and DNP3 (SCADA protocol). Examples include TCP/IP (transport layer), IPv4 and IPv6 (network layer), Fibre optics and Ethernet (physical layer).

Control Software: SCADA, PLC, and HMI software are also considered intangible critical assets. The vulnerabilities in software can cause detrimental effects on wind turbine operations if successfully exploited.

2) Automatic Identification

The automatic identification of critical assets in CPS is a complex problem, as many physical and logical subsystems are involved. Akbarzadeh et al. [9] propose an algorithm based It determines the criticality of the assets based on Closeness Centrality [CC], Tacit Input Centrality [TIC], and Tacit Output Centrality [TOC] to identify the critical assets. This is an effective algorithm to automate the critical asset identification process. However, considering the complexity of the CPS, it is not sufficient. We can identify certain assets based on Closeness centrality and the number of interconnections. However certain assets may not satisfy this criteria yet be performing critical functions. For instance, a servo drive in a wind turbine does not have many links and interconnections, yet its failure can lead to a wind turbine's malfunction. Liu et al. [23] have worked on the problem of automated identification of critical assets. Their approach constructs a network topology graph; nodes represent the assets, and edges denote logical or physical connections between them. The number of edges associated with a node indicates the criticality of the corresponding

asset. A high number of edges incident on a node make it critical. An asset's criticality rank depends on the number of edges associated with the node representing the asset. This algorithm considers interconnection information to determine the criticality. We plan to augment the algorithms [23] and [9]. In addition to the interconnections between the assets and closeness proximity, we will also consider the type of the assets to assess the criticality. The improved version of the algorithm is described below:

Each node will have a label indicating its criticality. The label can be initialized with information from the UML diagram or set to some initial value. The label of the node whose criticality is unknown can be computed based on the number of edges incident to it, and its distance from other critical nodes. This task will define parameters that will help identify the criticality of assets automatically.

Using the UML modeling discussed in IV-A, we construct nodes and edges using the Python library. The nodes represent assets, and the edges denote the interconnections between the assets. We also extract assets' attributes from the UML modeling to determine the critical assets. begi Next, we rank the critical assets based on the asset type and the total number of edges. The physical devices are labeled with the highest ranking, and the next ranking is assigned to the control devices as they directly control the physical devices. The other network devices are ranked based on the total number of input and output edges.

The proposed algorithm is limited to finding only cyber assets. The physical and intangible assets, such as communication protocols and software, are identified manually based on domain knowledge. The detailed algorithm implementation is out of the scope of this paper, and we discuss the detailed implementation of critical assets in the next paper.

C. Identification and Automatic Assessment of Vulnerabilities

In this process, we identify the vulnerabilities of each critical asset and provide a high-level algorithm for automatic vulnerability assessment.

We implemented a Python library to extract vulnerabilities from the vulnerability database for each critical asset. The Python code lists the vulnerability for each critical asset with its corresponding CVE and description. Example vulnerabilities produced by core are listed below.

PLC: Siemens S71500 contains vulnerabilities CVE-2020-15782 and CVE-2019-10943. CVE-2020-15782 allows attackers to write a protected memory location. CVE-2019-10943 allows modification of user code over the TCP port 102 such that the running code is different from the source code stored in the device memory.

Master/Save HMI: Siemens contain multiple vulnerabilities, such as CVE-2020-1578 and CVE-2020-15796, that allow an attacker remote access and cause DoS attacks through crafted HTTP requests without credentials.

Windows Server 2012 contains multiple vulnerabilities (CVE-2019-0575, CVE-2019-0578, CVE-2019-0582, amCVE-2019-0576) that allow remote code execution when the Windows Jet Database Engine improperly handles objects in memory.

Profinet contains vulnerability CVE-2019-19707, which can cause an attacker to execute a DoS attack by sending a legitimate Profinet packet over the network.

All vulnerabilities of any critical asset in a given CPS may not be exploited. The probability of vulnerability exploitation depends on the network topology, interconnection with the other critical assets, and current security posture. The goal of this task is to develop algorithms for automated vulnerability assessment. In the following, we define the high-level steps involved in evaluating whether a given vulnerability is applicable to our topology.

Step 1: Extract the vulnerabilities from the vulnerability databases for the critical asset.

Step 2: Extract pre and postconditions from the vulnerability description. A precondition is a minimum requirement to exploit a vulnerability, such as network access, authentication, etc. Post-condition is the state of the system after exploiting the vulnerability. We use Natural Language Processing (NLP) to extract pre and postconditions.

Step 3: Identify the possible access paths for the given critical asset from the network topology.

Step 4: Identify the entry point. Evaluate the pre and post-conditions for the entry point. If the precondition is met, the attackers can exploit the vulnerability in the system.

Step 5: If the postcondition is satisfied by exploiting the vulnerability, verify the next asset in the path for pre and postcondition.

Step 6: If the postcondition fails, then the next asset cannot be exploited, terminating the vulnerability assessment process for that asset.

Step 7: If the postcondition is met, then verify if the post-condition leads to the next adjacent asset. If it leads to the next adjacent asset, then vulnerability exploitation leads to the propagation of attack, and an attack path exists between the given critical asset and the next adjacent asset. We generate an attack path using the Python Plotly library to show the propagation of the attack.

D. Modeling the Cascading Effect of Attacks

We use System Dynamics (SD) tools to analyze an attack's cascading effect. System Dynamics (SD) is a mathematical modeling approach to design and analyze complex systems. Causal Loop Diagram (CLD) is an SD tool used to understand the impact of a variable on a complex system [21].

A CLD consists of four fundamental components: the variables, the links that connect them, the signs on the links, and the loop sign. The signs on the links show how the variables are interconnected. The signs of the loop show the loops' behavior, either reinforcing (R) or balancing (B). We consider **Wind Turbine Control Center** and **Wind Turbine** components from Figure 3 to demonstrate the cascading effect of an attack. Example CLD in Figure 3 demonstrates the cascading effect of a cyberattack on one asset on the entire system. The CLD consists of three loops: reinforcing loop A,

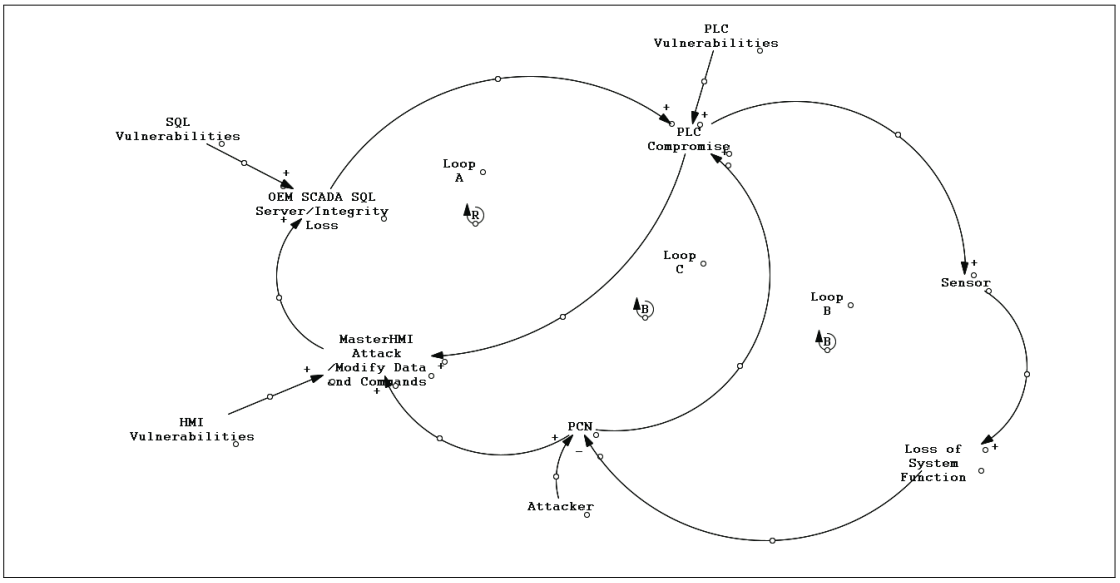

Fig. 3: Causal Loop Diagrams

balancing loop C, and balancing loop B. This section discusses the cascading effect of cyberattacks through loop B. "+" sign on the arrow connecting the assets represents reinforcing behavior, and "-" sign represents balancing behavior.

[Step 1: Attack on PCN Network {+} → HMI]: An attacker can tap the PCN by exploiting the vulnerabilities in "OEM Firewall," connecting the control devices to intercept the packets and extract information such as IP addresses, increasing the probability of an attack on the HMI by exploiting its vulnerabilities.

[Step 2: HMI {+} → OEM SQL SCADA Server] An increased probability of the HMI attack will increase the probability of the SQL server attack. Unintended operations and data modifications on the compromised HMI provide incorrect information to the SQL, causing an integrity attack on the "OEM SCADA Server". Additionally, SQL vulnerabilities, such as remote code execution, increase the probability of compromising the network and critical data.

[Step 3: OEM SCADA Server {+} → PLC] The attack probability on the "OEM SCADA Server" increases the probability of compromising PLC in two ways. The compromised critical data provides false information to the operators/users, who may provide false commands to the PLC. For example, in the case of a wind farm, incorrect pitch information provides false information to the SCADA operator. Based on the false information, the SCADA operator might provide a false command to the PLC to adjust the pitch, leading to detrimental consequences. Another possibility of compromising PLC is through an "OEM SCADA Server" compromised using injection attacks. An attacker can exploit the remote code execution vulnerability to gain access to the network. After accessing the network, the attacker can exploit the PLC vulnerability to send crafted TCP packets to the PLC.

[Step 4: PLC {+} → Sensors] The attack probability on the PLC leads to an increased probability of malfunctioning of wind-farm components, such as sensor failure, uncontrolled pitch, and voltage sag, leading to severe consequences, causing the shutdown of the wind-farm.

[Step 5: Sensors {-} → PCN] Shutting down the wind farm due to a cyberattack takes down the entire network. Network shutdown will reduce the probability of exploiting the PCN network, thus balancing the loop.

V. Phase 2: Risk Mitigation with Zero Trust Architecture

We extend the work of Gowdanakatte et al. [19] to implement the ZTA as a plug-and-play module to integrate the critical assets and their access points without modifying the original architecture. The critical assets are not directly accessible. All users requesting access to them have to go through the ZTA. We refer to incoming user requests as *request packet*. All authorized users with their user id (*UId*), the password (*Pwd*), the access level (*AccessLevel*), and the engineering workstations with their *DeviceId* must be pre-registered with the ZTA module in order to communicate with a critical asset within the given CPS.

A. Architecture of Zero Trust Module

Figure 4 represents the architecture of the ZTA module. The components of the ZTA modules are shown in orange rectangular boxes enclosed in a dotted orange outline. The blue rectangular box denotes the user requesting the resource, denoted by the green circle. The components of ZTA are described below.

Authentication Module consists of a TCP server socket listening to incoming user requests. It is responsible for authentication and handles the communication between the

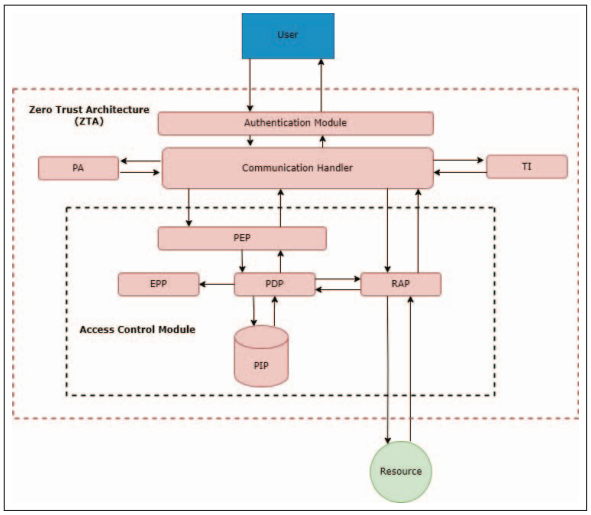

Fig. 4: Zero Trust Architecture

communication handler and the user.

Protocol Analyzer parses the *request packet* to extract the protocol header, the operation to be performed on the resource, and data related to the operation. It notifies the communication handler of the operation or that the corresponding data is invalid. Otherwise, it returns the operation and the data from the *request packet* to the communication handler. The protocol analyzer is specific to a given asset. It depends on the asset's protocol how it communicates with the other devices.

Threat Intelligence (TI) is a collection of Techniques, Tactics, and Procedures (TTP)s of the already known threats. The communication handler uses this information to verify if the *request packet* matches the pattern of already known threats.

Communication Handler (i) Communicates with the protocol analyzer for the deep packet inspection of the *request packet*. (ii) Communicates with the TI to verify if the *request packet* matches the pattern of already known threats. (iii) Forwards the *request packet* to the access control module for the access control verification. (iv) Establishes, maintains, and terminates the user and resource connection based on the decision computed by the access control module and the validation done by the protocol analyzer and threat intelligence.

Access Control Module enforces the least privilege per-request access for the requested resource through NIST NGAC-based Attribute Based Access Control (ABAC) [24]. The access control module comprises

1) **Policy Enforcement Point (PEP)** receives the *request packet* from the communication handler and forwards it to the Policy Decision Point (PDP).
2) **Policy Decision Point (PDP)** Extracts the policy from Policy Information Point (PIP), extracts the resource's status through Resource Access Point (RAP), computes the decision, and forwards the decision to the communication handler through the PEP. It also stores the record of incoming requests, time stamps, and the decision in the Event Processing Point (EPP).
3) **Policy Information Point (PIP)** stores the policy information for accessing critical resources. The PIP incorporates databases for user attributes, access policies, and geo-ip database for mapping IP addresses with locations.
4) **Resource Access Point (RAP)**The communication handler and PDP communicate with the resource through the RAP. The RAP is vendor specific to a given resource based on the communication protocol that the resource uses for communication.

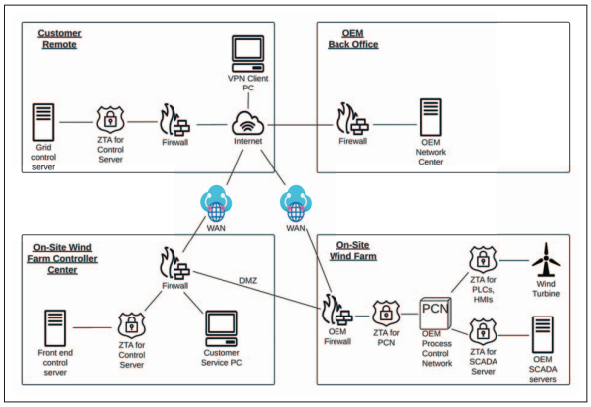

Fig. 5: Zero Trust Architecture for Wind Farm

B. Integration of ZTA with Wind farm Reference Architecture

We apply Zero Trust gateway between a critical asset, and it's access point. Figure 5 represents how the Zero Trust module can be installed between a critical asset and its access point in the reference wind farm architecture. 'ZTA for Control Server' between 'Grid Control Server' and the firewall in 'Customer Remote' station addresses vulnerabilities (CVE-2020-0618, CVE-2019-1332,CVE-2020-1455, CVE-2021-1636) in 'Grid Control Server'. 'ZTA for Control Server' between 'Front End Control Server' and the firewall in 'On Site Wind Farm Control Center' addresses the vulnerabilities (CVE-2020-0618, CVE-2019-1332, CVE-2020-1455, CVE-2021-1636) in 'Front End Control Server.' 'ZTA for PCN' in 'On-site Wind Farm' takes care of the access control vulnerabilities in industrial protocols (CVE-2020-25159, CVE-2020-2840) communicating through PCN. 'ABAC for SCADA Servers' addresses the access control vulnerabilities in the SCADA servers. 'ZTA for PLC and HMIs' addresses the access control vulnerabilities in the control devices(CVE-2020-15798,CVE-2021-37185) in the wind turbine.

C. Security Analysis with ZTA

We use Siemens PLC as a reference critical asset and demonstrate how the ZTA prevents a DoS attack on the PLC.

1) Example Policies

NGAC is a generic Attribute Based Access Control (ABAC) [25] architecture suitable for CPS and event driven applications. The NGAC model consists of basic elements and

relations. The basic elements comprise resources, users, operations, user attributes, and policy classes. NGAC expresses policy through assignment, association, prohibition, and obligation relations. [19].

For example, the PLC has NGAC policies as defined below.

- **Users** are the entities that request access to the resources.
- **Resources** are PLC components that need protection.
- **Environments** define the external conditions to users and resources needed for the access, such as location (IP address) and access time.
- **Operations** are actions a user can perform on the PLC.

A policy in the NGAC module is a tuple as:
$\langle\{userAttr\}, \{resourceAttr\}, \{envAttr\}, \{op\}\rangle$
where *userAttr*, *resourceAttr*, and *envAttr* denote the conditions on User Attributes, Resource Attributes, and Environment Attributes respectively, and *op* signifies operations. This policy states that *op* is allowed only when the *userAttr*, *resourceAttr*, and *envAttr* are satisfied. If any of the conditions are false, the access is denied.

For instance, a Communication Setup Policy (*CommSetup*) is permitted provided the user has access level *Operator*, *Engineer*, and *Administrator* with device *"4c174602"* and the time of access is in the interval 7:00-16:00 EST. This policy is expressed as:

$$\langle\{User.AccessLevel \in \{Operator, Engineer, Administrator\} \wedge User.Device = \text{“4c174602”}\}, \{True\}, \{Env.Time = 700 - 16:00 EST \wedge Env.Loc = \text{“}Org.local\text{”}\}, \{CommSetup\}\rangle$$

The first component in the tuple gives conditions on the user attributes. There are no explicit conditions on the resource attribute, so the second is "true". The third component signifies condition over environmental attributes: location and time. The last component denotes the allowable operation.

2) Threat Model

We developed a threat model for Siemens PLC in the wind turbine based on the work of Biham et al. [26]. We assume that (i) the attacker has the knowledge to attack Siemens PLC. (ii) The attacker is an outsider and has no authorized access to the wind turbine components. (iii) The attacker can access the OEM SCADA server through a compromised workstation.

Phase-1: Man-in-the-Middle attack – Interception of authenticated communication between the PLC and engineering workstation

The attacker performs the following operations:

1) Accesses the OEM SCADA Server through a compromised workstation. 2) Exploits the 'remote code execution vulnerability of the Server to access the PCN. 3) Exploits the vulnerabilities in the PCN network to access the Siemens PLC in the wind turbine. 4) Intercepts the PLC and the OEM SCADA Server communication through PCN. 5) Extracts the information on data required for establishing the communication with the PLC. This data depends on the specification of the PLC. We need the firmware version for the Siemens PLC to generate the communication request packet. 6) The attacker also extracts the function code and the related data of the operation.

Phase-2: Launching a DoS attack The attacker establishes the communication with the targeted PLC to cause an availability attack as follows: 1) Creates a communication request packet using the information obtained in Phase-1. 2) Sends the communication request TCP packet to the targeted PLC's IP address and the TCP port. 3) The PLC establishes communication with the attacker through the compromised workstation. 4) The attacker then sends a crafted TCP packet with the modified function code and data to the TCP port of the PLC to cause a DoS attack.

3) Prevention of DoS attack

Case 1: Access from Unauthorized Engineering Workstation The attacker first requests to establish communication with the PLC through an unauthorized workstation. The authentication module receives the *request packet* and prompts the attacker to enter the *UId* and the *Pwd*. The authentication fails as the attacker is not a registered user, and the authentication module disconnects from the attacker's engineering workstation. The authentication is successful if the attacker has access to compromised credentials of a registered user. In such a case, the authentication module extracts the *DeviceId* from the engineering workstation and forwards the *request packet*, *UId*, and *DeviceId* to the communication handler. The communication handler verifies the *request packet* with PA and TI for deep packet inspection and the matching attack patterns. Both PA and TI validate the *request packet* as it does not contain any malicious function code or existing attack patterns. The communication handler forwards the *request packet*, *DeviceId*, and the *UId* to the access control module for access control verification. The PDP extracts the policy from PIP and computes the decision. It denies the access as the attacker's *DeviceId* is not registered in the PIP. Hence it sends the decision *Deny* to the communication handler. The communication handler requests the authentication module to disconnect from the attacker's engineering workstation.

Case 2: Access from a Compromised Engineering Workstation The attacker first requests to establish communication with the PLC. The authentication module receives the *request packet* and prompts the attacker to enter the *UId* and the *Pwd*. The authentication of the attacker fails as the attacker is not a registered user, and the authentication module disconnects from the compromised SCADA server. The authentication succeeds if attacker has the compromised credentials of a registered user. The authentication module then extracts the *DeviceId* from the SCADA server and forwards the *request packet*, *UId*, and *DeviceId* to the communication handler. The communication handler verifies the *request packet* with PA and TI for deep packet inspection and the matching attack patterns. Both PA and TI will validate the *request packet* if it does not contain any malicious function code or existing attack patterns. The communication handler forwards the *request packet*, *DeviceId*, and the *UId* to the access control module for verification. The PDP extracts the policy from PIP and computes the decision. It approves the access as the Server's *DeviceId* is registered in the PIP. Hence it sends the decision *Approve* to the communication handler.

The communication handler forwards the *request packet* to the PLC through the RAP to establish the connection. The PLC receives the connection requests and establishes the connection with the attacker. The attacker sends the crafted TCP packets with the invalid function code to the PLC through the ZTA. The authentication module receives the crafted TCP packets and requests for the authentication. If the authentication is successful, it forwards the TCP packets to the communication handler The communication handler verifies the TCP packet with the PA and TI. The PA module rejects the TCP packets with the invalid function code. The communication handler requests that the authentication module disconnect from the SCADA server and reject the TCP packets.

VI. Conclusion and Future Work

The lack of identification of risks and integration of security during the design phase is the fundamental cause of increased cyber attacks on Cyber Physical Systems (CPS). Our method provides a holistic model-based risk identification and security integration during the design phase.

Many research is left to be done. One issue is understanding what effects does temporal factors have on the risk assessment process. Second issue pertains to automated identification of new threats, and providing a visual framework that displays such threats and their impact.

Acknowledgement

This work was supported in part by funding from NSF under Award Numbers ATD 2123761, CNS 1822118, ARL, Statnett, AMI, NewPush, and Cyber Risk Research and funding from NIST under Award Number 60NANB23D152.

References

[1] Unknown, "Power grids under attack," https://semiengineering.com/power-grids-under-attack/.

[2] I. T. R. Team, "Industroyer2 Malware Targeting Ukrainian Energy Company," https://www.ironnet.com/blog/industroyer2-malware-targeting-ukrainian-energy-company.

[3] T. W. S. Journal, "European Wind-Energy Sector Hit in Wave of Hacks," https://www.wsj.com/articles/european-wind-energy-sector-hit-in-wave-of-hacks-11650879000.

[4] N. GROUP, "Redecho Targets the Indian Power Sector," https://www.recordedfuture.com/redecho-targeting-indian-power-sector.

[5] Insikt, "Attacks on uta wind and solar utility," https://www/recordedfuture.com/redecho-targeting-indian-power-sector, Tech. Rep., 2021.

[6] B. Ankica, R. Ivan, S. Dušan, A. Mustafa, A.Rima, W. Li, M.Hana, E. Raheleh, C. Moharram, B. Dominique, N.Oksana, and C.Antonio, "Multi-paradigm modeling for cyber–physical systems: A systematic mapping review," *Journal of Systems and Software*, vol. 183, p. 111081, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121221001783

[7] NIST, "Nist risk management framework," https://csrc.nist.gov/projects/risk-management/about-rmf, Tech. Rep., 2016.

[8] A. Akbarzadeh and S. Katsikas, "Dependency-Based Security Risk Assessment for Cyber-Physical Systems." International Journal of Information Security, 2023.

[9] ——, "Identifying critical components in large scale cyber-physical systems." 42nd International Conference on Software Engineering, 2020.

[10] Y. Peng, T. Lu, J. Liu, Y. Gao, X. Guo, and F. Xie, "Cyber-physical system risk assessment," in *2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2013, pp. 442–447.

[11] Y. Kawanishi, H. Nishihara, D. Souma, H. Yoshida, and Y. Hata, "A study on quantitative risk assessment methods in security design for industrial control systems," in *22018 IEEE 16th Intl Conf on Dependable Autonomic and Secure Computing*, 2018.

[12] I.Semertzis, R. V. Subramaniam, A. Ştefanov, F. Fransen, and P. Palensky, "Quantitative risk assessment of cyber attacks on cyber-physical systems using attack graphs," in *2022 10th Workshop on Modelling and Simulation of Cyber-Physical Energy Systems*, 2022, pp. 1–6.

[13] X. Zhang and D. Zhang, "Quantitative risk assessment of cyber-physical power system using bayesian based on petri net," in *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, 2018, pp. 988–992.

[14] A. Sundararajan, A. Chavan, D.Saleem, and A.Sarwat, "A survey of protocol-level challenges and solutions for distributed energy resource cyber-physical security," *Energies*, 2018. [Online]. Available: https://www.mdpi.com/1996-1073/11/9/2360

[15] T. Krause, R. Ernst, B. Klaer, I. Hacker, and M. Henze, "Cybersecurity in power grids: Challenges and opportunities," *Sensors*, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/18/6225

[16] I. Zografopoulos, J. Ospina, X. Liu, and C. Konstantinou, "Cyber-physical energy systems security: Threat modeling, risk assessment, resources, metrics, and case studies," *IEEE Access*, pp. 29 775–29 818, 2021.

[17] Y.Zhang, Y.Xiang, and L.Wang, "Power system reliability assessment incorporating cyber attacks against wind farm energy management systems," *IEEE Transactions on Smart Grid*, pp. 2343–2357, 2017.

[18] Z.Gunduz and R.Das, "Analysis of cyberattacks on smart grid applications," in *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, 2018, pp. 1–5.

[19] S. Gowdanakatte, I. Ray, and S. H. Houmb, "Attribute based access control model for protecting programmable logic controllers," in *SaT-CPS*. New York, NY, USA: ACM, 2022, p. 47–56.

[20] L. Ordinez, G. Eggly, M. Micheletto, and R. Santos, "Using uml for learning how to design and model cyber-physical systems," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, pp. 50–60, 2020.

[21] M. Saeri, M. Lotfi, and M. Mazidi, "A causal loop diagram to analyze various long-term effects of pv integration into power systems," in *2019 27th Iranian Conference on Electrical Engineering (ICEE)*, 2019, pp. 852–855.

[22] Python, "Plantuml 0.3.0," https://pypi.org/project/plantuml/, 2019.

[23] C. Liu, Y. Alrowaili, N. Saxena, and C. Konstantinou, "Cyber risks to critical smart grid assets of industrial control systems," *Energies*, 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/17/5501

[24] D. Ferraiolo, R. Chandramouli, D. Kuhn., and V. Hu, "Extensible Access Control Markup Language (XACML) and Next Generation Access Control (NGAC)," in *ACM International Workshop*. Association for Computing Machinery, 2016, pp. 13–24.

[25] A. N. S. Institute, "Information technology: Next Generation Access Control (NGAC)," ANSI, Tech. Rep., 2020.

[26] E. Biham, S. Bitan, A. Carmel, A. Dankner, U. Malin, and A. Wool, "Rogue Engineering Station Attacks on Simatic S7 PLCs," https://i.blackhat.com/USA-19/Thursday/us-19-Bitan-Rogue7-Rogue-Engineering-Station-Attacks-On-S7-Simatic-PLCs.pdf/, 2019.